NEC Personal Computer

PC-8801 mkII
USER'S MANUAL

PC-8801MK2-UM
PTS-247B

PC-8801 mkII

USER'S MANUAL

はじめに

このたび，PC-8801MKⅡをお買い求めいただきましてありがとうございます。

日本電気は，PC-8000シリーズを始め数多くのパーソナルコンピュータを発売してまいりました。各シリーズとも，その特徴を生かし，ホビーからビジネスまで幅広く利用されております。

PC-8801MKⅡは，最新の技術をもとに，PC-8801をコンパクト化し，ローコストマシーンとして製品化したものです。

注意事項

◎電源に関する御注意

(1) 電源スイッチを一度「OFF」にしたときは5秒以上経ってから「ON」にして下さい。電源スイッチが「ON」の状態で電源プラグを抜いたときも同様に5秒以上経ってから電源プラグを差し込んで下さい。

(2) 電源は必ずAC100V(50Hzあるいは60Hz)を使用して下さい。

(3) 電源コードを抜き差しする場合は必ずプラグのところを持って行って下さい。

◎ 保管および使用環境に関するご注意

(1) 本機は温度上昇を防ぐため,ケースに通風孔が開けてありますので,通風孔をふさいだり,風通しの悪い場所でのご使用をさけてください。又本機を極端な高温下や低温下または温度変化の激しい場所で保管及び使用することはさけて下さい。

(2) 本機を直射日光の当る場所や発熱をする器具の近くで保管および使用することはさけてください。

(3) 本機を極端に湿気の多い場所や,ほこりの多い場所で保管および使用することはさけて下さい。

(4) 本機は精密な電子部品でできていますので,衝撃を加えたり,衝撃,振動の加わる場所での保管および使用することはさけて下さい。

(5) 本機の内部に水や液状のもの,金属類が入った状態でご使用になりますと危険ですので,異物が入らないようご注意下さい。

(6) 薬品の雰囲気中や薬品に触れる場所での保管および使用することは避けて下さい。

(7) 本機を解体した状態で保管および使用することは,故障や感電の原因になりますのでおやめ下さい。

(8) 本機の上に重い物を置いた状態で保管および使用することはさけて下さい。

(9) ラジオやテレビなどのすぐそばで使用しますと,ラジオやテレビに

雑音が入ることがあります。また，強い磁界を発生する装置などが近くにありますと，逆に本機に雑音が入ってくることがあります。このような場合は離してご使用ください。

⑽ PC-8801MKⅡは，本体後部にアース端子が備わっています。アース線が配線されている場所ではアースを取ることをお勧めします。

◎その他

(1) 本機の汚れはやわらかい布に水または洗剤を含ませて軽くふいて下さい。ベンジン，シンナーなど(揮発性のもの)や薬品を用いてふいたりしますと変形や変色の原因になることがあります。また殺虫剤などをかけた場合でも変形や変色の原因になることがありますので，ご注意下さい。

◎異常，故障の場合

(1) 故障や異常(臭いがしたり，過熱していたり)に気付いたときは，直ちに電源コードのプラグを抜いて，お買い求めの販売店あるいはもよりのBit-INNにご相談下さい。

PC-8801MKII

ユーザーズ・マニュアル

目　次

第1章

第1章
概　　説

この章では，PC-8801MKⅡの付属品の使い方と，このマニュアルの見方について説明します。

1.1　付属品の使い方

PC-8801MKⅡには，次のような付属品が添付されています。

・キーボード(１個)
・ケーブル(３本)
・マニュアル(３冊)
・ミニフロッピィディスク(2枚)または，カセットテープ(１本)

これらの使い方は，次の表のとおりです。

名　称		外　見	用　途
キーボード			本体に接続して，文字などの入力に使います。 4.3(1)参照
ケーブル	電源ケーブル		本体に，AC100Vの電源を供給するケーブルです。

名　称		外　見	用　途
ケーブル	モノクロディスプレイ用ケーブル（PC-8092）		本体と専用のモノクロディスプレイを接続するケーブルです。 4.3(2)参照
	カセットテープレコーダ用ケーブル（PC-8093）		本体と，カセットテープレコーダを接続するケーブルです。 4.3(5)参照
マニュアル	PC-8801MKIIユーザーズマニュアル		周辺装置との接続，基本的な操作法などについて，ぜひ知っておいていただきたいことが書かれています。
	BASICリファレンスマニュアル		PC-8801MKIIのプログラミング言語であるN-BASIC，N88-BASICの文法書です。
	BASICクイックリファレンス		「BASICリファレンスマニュアル」をコンパクトにまとめた小冊子です。
ミニフロッピィディスク	N88DISK-BASICシステムディスク	*	N88-BASICのディスク版であるN88DISK-BASICのディスクコードおよび，ユーティリティプログラムが入っています。 付録5.1参照
	N88DISK-BASICゲームディスク	*	N88DISK-BASIC上で動作するゲームが入っています。 付録5.1参照

名　　称	外　見	用　　　途
カセットテープ	＊＊	PC-8801MKⅡを紹介するためのデモンストレーションプログラムが入っています。 付録5.2参照

注　意

機種によって添付品が一部異なります。

＊）　PC-8801MKⅡ model 20, model 30のみ添付。

＊＊）　PC-8801MKⅡ model 10のみ添付。

1.2　このマニュアルの内容と使い方

このマニュアルは，梱包箱をあけたあとはじめに読んでいただきたいマニュアルです。第1章から順に読んでいただければ，周辺装置の接続のしかた，起動方法，基本的な操作方法，PC-8801MKⅡの機能などがご理解いただけると思います。

基本的な操作や機能を前半で,応用的な事柄を後半で説明してあります。

・第2章～第8章

PC-8801MKⅡの設置方法，起動方法，操作方法などについて説明してあります。必ず読んでください。

・第9章～第15章

応用的な，PC-8801MKⅡの使用法や，機能の拡張についての説明です。BASICを使ってプログラムを作るときや，ターミナルモードとして使うときに参照して下さい。

各章の使い方は，次のとおりです。

・システムを組むまで……………………第1章，第2章，第3章，第4章，第8章
・起動と基本操作…………………………第5章，第6章，第7章
・BASICでプログラムを作るとき ………第9章，第10章，第11章，第12章
・機械語モニタの使い方…………………第13章
・ターミナルモードの使い方……………第14章
・拡張命令の使い方………………………第15章

1.3　このマニュアルで使われている記号

このマニュアルでは，わかりやすく説明するために，次の記号を使っています。

記号	説明
⏎	リターンキーの入力を表します。
Note	ぜひ覚えておいていただきたい事柄が書かれています。
使用例	使用例などを表します。
注意	特に注意が必要な事柄であることを表します。
参照	参照していただきたい関連項目を表します。
解説	用語などについて解説しています。
PC-8801とくらべて	PC-8801MKⅡとPC-8801をくらべてどこがどうちがうかを表します。
XXXXH	数字にHをつけて16進数を表します。
△	スペースを表します。

第2章

第2章
PC-8801MKIIの特長

2.1　ハードウェアの特長

PC-8801MKⅡのハードウェアの特長として以下のようなものがあります。

① PC-8801とのハード的な互換性

・PC-8801MKⅡ＝PC-8801＋PC-80S31＋漢字ROM

・PC-8800シリーズの周辺装置が使えます。

② CPUにμPD780C-1を2ケ使用

・メインCPU…メインメモリのコントロール，画面制御，キー制御，割込みコントロールなど

・サブCPU……5"FDDのコントロール

③ 202Kバイトのメモリを標準実装

・ROM ……N-BASICおよびモニタ		32Kバイト
	N_{88}-BASIC	40Kバイト
	5"FDDの制御	2Kバイト
・RAM ……メインメモリ		64Kバイト
	グラフィック用VRAM	48Kバイト
	サブCPU用メモリ	16Kバイト

④ 漢字ROMを内蔵しており，日本語の表示が可能

⑤ 強力なグラフィックス機能

・640×200ドット(8色カラー1ページ又は，白黒3ページ)

・640×400ドット(白黒1ページ)

・テキスト画面とグラフィックス画面の合成表示が簡単にできます。

・パレット機能により多彩なカラー表示が可能です。

⑥ ミニ・フロッピィディスク(内蔵)により，大量のデータを高速に処

理することが可能

・model 20には1ドライブ, model 30には2ドライブのミニフロッピィディスクが内蔵されています。

⑦ 柔軟な拡張性

・拡張用スロットにより種々なインタフェース・ボードを接続することができます。

⑧ 豊富なインタフェース機能

・オーディオ・カセット・インタフェース

・プリンタ・インタフェース(セントロニクス社仕様に準拠)

・シリアル・インタフェース(RS-232C規格に準拠)

・汎用入・出力ポート(バーコード・リーダ接続可能)

⑨ 場所をとらないコンパクトなボディーで，縦置き，横置きが可能

PC-8801とくらべて

PC-8801とPC-8801MKⅡの相異点は次のとおりです。

PC-8801	PC-8801MKⅡ
ミニフロッピィディスク・インタフェースのみ内蔵	ミニフロッピィディスク・ドライブ内蔵（model 20, model 30）
内蔵スピーカーの音量調節不可	内蔵スピーカーの音量調節可
拡張用スロット　4スロット	拡張用スロット　3スロット
汎用I/Oポートとカセットインタフェースのコネクタが共用	カセットインタフェース用コネクタと汎用I/Oポート用コネクタが独立
漢字ROMボードがオプション	漢字ROMを内蔵
メモリ・ウェイト用ディップスイッチなし	メモリ・ウェイト用ディップスイッチあり
ボーダーカラーの機能あり	ボーダーカラーの機能なし

2.2　ソフトウェアの特長

PC-8801MKⅡには，BASICの他に，便利なユーティリティプログラムなどが付属しており，手にしたその日から，様々な用途に使えるようになっています。

① 他のPCシリーズと互換性のあるBASIC

・N-BASICとN88-BASICの２つのBASICが使え，現在数多く流通しているPC-8001, PC-8801用のソフトウェアのほとんどがそのまま利用できます。*

② 拡張命令パッケージ

・N88-BASICに拡張命令パッケージを付加することにより，タートルグラフィック機能や，サウンド機能が追加されます。

③ その他のアプリケーションソフトウェア**

・その他に，パソコンの使い方を教えてくれるチューター(パソコン家庭教師プログラム)や，アドベンチャーゲームなどが付いています。

*）PC-8801用のソフトウェアで，PC-8801MKⅡでは一部同じように動作しないものがあります。

**）model 20, model 30のみ。

第3章

第3章
システム構成

PC-8801MKⅡは，本体だけでは何の役にもたちません。ステレオでもアンプにプレーヤやスピーカをつながなければならないように，PC-8801MKⅡにも周辺装置を接続してはじめて，使えるようになるのです。このように，必要な周辺装置を接続して，できるセットをシステムといいます。

上図に一般的な本体と周辺装置の関係を示しました。本体にいろいろな周辺装置が接続されています。本体(CPUともよばれます)にあたるのがPC-8801MKⅡです。

この章では，PC-8801MKⅡのシステム構成例，ハードウェア構成図，主な周辺装置について紹介します。

3．1　システム構成例

PC-8801MKⅡは，数多くの周辺装置を組み合わせることにより，いろいろな用途に合わせて各種のシステムを選択できます。ここにその一例として，3種類の例を紹介します。なお，システム構成につきましては，もよりのBit-INNもしくは，お買い上げの販売店で御相談を承っています。

(1)　ホビー向けのシステム構成

PC-8054
PC-8091
PC-8093
PC-8801MKII
PC-DR311

本体にCRTディスプレイを接続したシステム構成です。ホビー向けのソフトは，ほとんどがカラーグラフィックの効果を利用していますので，CRTディスプレイとしてPC-8054を接続します。ただし，この場合は640×400モードは使えません。

(2) ワードプロセッサ向けのシステム構成例

PC-8851
PC-8092
PC-8894
PC-8825
PC-8801MKII

このシステムは，PC-8801MKⅡに専用高解像度モノクロディスプレイ(PC-8851)と，熱転写漢字プリンタ(PC-8825)を接続したもので，日本語ワードプロセッサとして使用する場合に適しています。

(3)　ビジネス向けのシステム構成例

PC-KD551	
PC-8091	
	PC-8882(1)
	PC-8881(1)
PC-8801MKII	PC-PR201
PC-8894	

このシステムは，ビジネス向けの構成例で，専用高解像カラーディスプレイ(PC-KD551)，日本語シリアルプリンタ(PC-PR201)，８インチフロッピィディスク・ユニット(PC8881(1)，PC8882(1))を接続したものです。

このシステムでは，標準フロッピィディスクドライブが４台接続されていますので，本体のミニフロッピィドライブ以外に4Mバイトの容量の外部メモリをもつことができます。また高速で鮮明な漢字プリンタも接続されており，大規模データ処理や漢字処理などのビジネスマシーンとしての機能を発揮できます。

PC-8881(1)，PC-8882(1)は，本体右側設置用です。左側設置には，PC-8881, PC-8882をお使いください。

3．2　主な周辺装置

PC-8801MKⅡに接続できる周辺装置の中で主だった物の外観とその機能を紹介します。

PC-8851　14インチ・モノクロ(専用高解像度)ディスプレイ

横幅：340mm　　高さ　　：332mm
奥行：340mm　　消費電力：60W(MAX)
重量：10kg

・水平周波数24.8kHzの採用により，最大640×400ドットの高分解能を実現しています。

PC-KD551　14インチ・カラー(専用高解像度)ディスプレイ

横幅：364mm　　高　さ：346mm
奥行：378mm　　消費電力：70W(MAX)
重量：13.0kg

・水平周波数24.8kHzの採用により，最大640×400ドットのカラー表示を実現しています。
・本体との接続ケーブルが添付されています。

注　意

PC-8851, PC-KD551は，PC-8801およびPC-8801MKⅡ専用のディスプレイです。PC-8001MKⅡやPC-6001MKⅡなどに接続して使用することはできません。

PC-8824　熱転写プリンタ
PC-8825　熱転写漢字プリンタ

横幅：437mm　　高　さ：119mm
奥行：275mm　　消費電力：40W(MAX)
重量：13.5kg

・PC-8824/PC-8825は，24ドット・サーマルヘッドを採用した高品質印字プリンタです。
・熱転写方式の印字により，印字音が非常に小さい「低騒音タイプ」です。
・PC-8824にオプションの漢字ROMボードを実装することにより，PC-8825と同じ機能になり，プリンタへ漢字コードを送るだけで，簡単に漢字を印字することができます。
・PC-8824/PC-8825には，プリンタケーブル(PC-8894)熱転写リボン(PC-8824-01)，普通ロール紙（PC-8824-P1）が添付されています。

PC-PR201　日本語シリアルプリンタ

横幅：570mm	高　　さ：125mm
奥行：335mm	消費電力： 70W
重量：9.5kg	

・PC-PR201は，24ピンプリントヘッドを使ったインパクトタイプの漢字プリンタです。
・用紙幅5インチ〜16インチのファンフォールド紙，カットシートが使用でき，ビジネス用途に適しています。
・JIS第1水準漢字を標準実装，また第2水準漢字ROMボードをオプションで装備可能となっています。
・オートローディング機能により用紙のセットが容易です。
・プリンタケーブル(PC-8894)，シートガイドが添付されています。

PC-80S32　拡張用ミニディスクユニット

横幅：154mm　　高　さ：195mm

奥行：338mm　　消費電力：25W

重量：6.5kg

・PC-80S32は，PC-8801MKⅡに増設可能な，両面倍密度拡張用ミニディスクユニットです。

・薄型ドライブを2台装備しており約640Kバイトの記憶容量をもちます。

・PC-80S32には，専用ケーブルが添付されています。

注　意

PC-80S32は，本体内にミニフロッピィディスクドライブが2基実装されている場合のみ使用できます。

PC-8881(1)　8インチ・標準フロッピィディスクユニット
PC-8882(1)　拡張用8インチ・標準フロッピィディスクユニット

横幅：194mm　　高　さ：250mm(ゴム足は含まず)
奥行：360mm　　消費電力：60W
重量：11.9kg

・PC-8881(1)は，8インチ薄型ドライブを2台実装していて，合計2Mバイトの記憶容量を持っています。さらに，PC-8882(1)と接続することにより，4Mバイトまで拡張することができます。

・DMA転送方式を採用していますから，高速でデータ転送を行うことができます。

・PC-8881(1)には，FDCインタフェースボードと接続ケーブルが付いています。また，PC-8882(1)には，PC-8881(1)との接続ケーブルが付いています。

・PC-8881(1)，PC-8882(1)は，本体右側設置用です。左側設置には，PC-8881，PC-8882をお使いください。

第4章

第4章
本体と周辺装置との接続

4.1　本　　体

(1)　外観および各部の名称

PC-8801MKⅡは，本体とキーボードが分離しているセパレートタイプで，本体とキーボードの間は，カールケーブルで結ばれています。

各部の名称は，次のとおりです。

ミニフロッピィ
ディスクユニット
本体
キーボードコネクタ
電源スイッチ
キーボード
コネクタ
ファンクションキー
電源表示用LED
BASIC表示用LED
扉
文字キー
および特殊キー
カーソル移動キー
キーボード
テンキー

本体正面およびキーボード

拡張用スロット
サービスコンセント
スロット1
スロット2
スロット3
電源
プリンタ用コネクタ
汎用I/Oポート
コネクタ
拡張ミニフロッピィディスク
ユニット用コネクタ
RS-232Cコネクタ
カラーディスプレイ用コネクタ
モノクロディスプレイ用コネクタ
オーディオ
カセット用コネクタ

本体背面

ブザー音量調整ボリューム
ディップスイッチ1
ディップスイッチ2
ジャンパースイッチ
リセットスイッチ

本体前面の扉を開けた図

(2) ディップスイッチ

PC-8801MKⅡの前面右下の扉を開けると向って左側に16個のディップスイッチがあります。その機能は次のとおりです。

SW1　　　SW2

ON
1 2 3 4 5 6 7 8

ON
1 2 3 4 5 6 7 8

（初期状態）

SW1	1	起動時のBASICモード	ON	N-BASIC
			OFF	N88-BASIC
	2	起動時のモードの選択(N88-BASICモードで有効)	ON	ターミナルモード
			OFF	BASIC
	3	起動時の1行あたりの文字数(N88-BASICモードで有効)	ON	80文字/行
			OFF	40文字/行
	4	起動時の1画面あたりの行数(N88-BASICモードで有効)	ON	25行/画面
			OFF	20行/画面
	5	Sパラメータ	ON	Sパラメータ有効
			OFF	Sパラメータ無効
	6	DEL コード受信時動作	ON	DELコードを処理する
			OFF	DELコードを無視する
	7	メモリウェイト	ON	1 WAIT動作をする
			OFF	通常
	8	バーコードリーダおよびCMD SINGの選択	ON	バーコードリーダ使用
			OFF	CMD SING使用
SW2	1	パリティチェック	ON	パリティ有り
			OFF	パリティ無し
	2	パリティ指定	ON	偶数(EVEN)パリティ
			OFF	奇数(ODD)パリティ
	3	データビット長	ON	8ビット
			OFF	7ビット
	4	ストップビット長	ON	ストップビット=2
			OFF	ストップビット=1
	5	Xパラメータ	ON	Xパラメータ有効
			OFF	Xパラメータ無効
	6	通信方式	ON	半二重
			OFF	全二重
	7	内蔵ミニフロッピィディスクドライブ	ON	あり
			OFF	なし
	8	未使用		

このスイッチのON, OFFはシャープ・ペンシルの先のようなもので上下に動かしてください。下向きがOFF，上向きがONです。

PC-8801とくらべて

PC-8801MKⅡでは，SW1の7，8およびSW2の7が追加されています。それ以外は，PC-8801と同じです。

(3) ジャンパースイッチ

PC-8801MKⅡの前面右下の扉を開けると，向って右側に15個のジャンパースイッチがあります。これは，RS-232Cのボーレートおよびディスプレイモードを設定するものです。

ジャンパースイッチは，全部で3つのブロックに分かれており，各ブロックに一個ずつソケットが付いています。一つのブロック中で2個以上のソケットをさすことはできませんから注意してください。

RS-232C　CRT

1 2 3 4 5 6 7 8　S H　M T（初期状態）

ボーレート切換（RS-232C）	1	75ボー
	2	150ボー
	3	300ボー
	4	600ボー
	5	1200ボー
	6	2400ボー
	7	4800ボー
	8	9600ボー
グラフィックでのタテの最大表示ドット数	S	標準ディスプレイ使用
	H	専用ディスプレイ使用
白黒モニタへの出力画面	M	テキスト＋グラフィック1，2，3
	T	テキスト

PC-8801とくらべて

PC-8801MKⅡでは，RS-232Cは内部同期だけになり，外部同期，内部同期の選択はできなくなっています。また，白黒モニタへの出力で，グラフィック画面の選択ができません。

4.2　設置に関する注意

(1)　設置場所

PC-8801MKⅡ及び周辺装置は，電子部品でできていますので，温度変化の激しい場所，極端な高，低温下，湿気やほこりの多い場所へ設置することは避けて下さい。

詳しくは，目次の前にある「保管及び使用環境に関するご注意」を参照して下さい。

(2)　周辺装置との接続に関する注意

PC-8801MKⅡには，PC-8800シリーズをはじめ，PC-8000シリーズやPC-6000シリーズの周辺装置など，数多くの周辺装置を接続することができます。

PC-8801MKⅡと各周辺装置を接続するときは，次の事柄を必ず守ってください。

(i)　PC-8801MKⅡと周辺装置を接続したり，切り離したりする場合は，必ずすべての装置の電源を「OFF」にした状態で行ってください。電源が「ON」になったままで，接続や切り離しを行うと，故障の原因となる場合がありますから十分注意してください。

(ii)　PC-8801MKⅡと各周辺装置との接続は，「4.3周辺装置との接続」で指定してあるケーブルを使用してください。使用するケーブルがわからない場合は，お買い求めの販売店もしくはもよりのBit-INNに御相談ください。

(iii)　各装置のソケットに矢印(▼印)がついている場合は，ケーブルのコネクタの矢印と一致するように接続してください。逆に接続しますと，誤動作や故障の原因となりますから注意してください。

(iv)　周辺装置は，ケーブルが引っぱられないような位置に設置してください。ケーブルが強く張りつめていたり，ケーブルの上に物を載せているような状態での使用は避けてください。

(v)　コネクタがきちんと差し込まれていないと，動作をしなかったり，あるいは誤動作を起こしたりします。コネクタは確実に取り付けてください。

周辺装置の接続や切り離しは，電源を「OFF」にしてから行いましょう！

(3) PC-8801MKⅡは，縦置き，横置きどちらの状態でも使えます。縦置きの場合は，足の付いている方を下にして，足は伸ばした状態でお使いください。

足を伸ばすには，コインなどを使って，ネジをゆるめてください。伸ばしたあとは，再びネジをしめておきます。

4.3　周辺装置との接続

(1)　キーボード

PC-8801MKⅡのキーボードは，本体前面の左下または，本体右側のキーボードコネクタに接続します。どちらに接続しても動作は同じですので，使い易い方にしてください。

注　意

キーボードは，片方のコネクタのみに接続し，残りのコネクタには，必ずカバーを付けるようにしてください。また接続する際

は，コネクタの向きに十分注意してください。

(2) ディスプレイ

PC-8801MKⅡには，PC-8853N, PC-KD551, PC-8851など〈専用高解像度ディスプレイ〉をはじめとして，PC-8054, PC-8050Nなどの高解像度ディスプレイや家庭用テレビなど〈標準ディスプレイ〉も接続できます。

〈専用高解像度ディスプレイ〉を使用する場合は，本体前面のジャンパースイッチを切り替える必要があります。この切り替えは，カラーディスプレイ用コネクタとモノクロディスプレイ用コネクタの両方を切り替えます。ですから，カラーディスプレイとモノクロディスプレイを両方同時に使用する場合は，両方とも〈専用高解像度ディスプレイ〉にするか，〈標準ディスプレイ〉にする必要があり，それに合わせてPC-8801MKⅡ本体前面のジャンパースイッチを設定します。

注　意

ジャンパースイッチを切り替える場合も，PC-8801MKⅡおよび，各周辺装置の電源は「OFF」にしてください。

〈標準ディスプレイ〉使用時(Standard display)

S H

〈専用高解像度ディスプレイ〉使用時(High scan. display)

S H

例）PC-8841, PC-8851, PC-8853N, PC-KD551

参　考

〈専用高解像度ディスプレイ〉は，水平周波数24.8kHzのディスプレイです。これに対し，〈標準ディスプレイ〉は，水平周波数が15.7kHzとなっています。

次にモノクロディスプレイの画面選択について説明します。PC-8801MKⅡでは，モノクロディスプレイ用コネクタに接続したディスプレイに出力する画面を本体前面のジャンパー・スイッチで選択することができます。

(i)　テキスト画面とグラフィック画面のMix画面

M　T

(ii)　テキスト画面だけ

M　T

参　照

テキスト画面とグラフィック画面については，「第８章ディスプレイモード」を参照して下さい。

PC-8801MKⅡと，カラーディスプレイを接続する場合には，専用ケーブルPC-8091(ディスプレイに添付)を使用します。

本体背面

カラーディスプレイ

また，モノクロディスプレイと接続する場合には，専用ケーブルPC-8092(本体に添付)を使用します。

家庭用テレビを接続する場合は，PC-8044K家庭用カラーアダプタを使用します。しかし，VTR用モニターテレビなどでは使用できないものもあります。詳しくは，お近くのBit-INNもしくはお買い上げの販売店で御相談ください。

注　意

PC-8801MKⅡの上にディスプレイをのせて御使用になる場合，ディスプレイは正面にむけてお使いください。

(3) プリンタ

PC-8801MKⅡには，PC-PR201をはじめ，セントロニクス規格に準拠したプリンタを接続することができます。接続には，専用ケーブルPC-8894を使用します。

専用ケーブルPC-8894は，PC-PR201, PC-8825などには添付されていますが，PC-8023-Cなど一部添付されていないものがあります。「PC-8801MKⅡ用プリンタケーブル」として購入してください。

(4) フロッピィディスクユニット

PC-8801MKⅡには，最高，標準フロッピィディスクを4ドライブ(PC-8881/PC-8881(1)とPC-8882/PC-8882(1))とミニフロッピィディスクを4ドライブ(本体内蔵とPC-80S32)まで接続して使用できます。

PC-80S32は，本体に2ドライブ実装されているときに，また，PC-8882/PC-8882(1)は，PC-8881/PC-8881(1)が接続されているときだけ増設することができます。

標準フロッピィディスクユニットを接続する場合は，PC-8801MKⅡ本体背面のスロット3に標準フロッピィディスクユニット用インタフェースボードをさし，それとPC-8881/PC-8881(1)か，PC-8882/PC-8882(1)を専用ケーブルで接続します。インタフェースボードおよ

び専用ケーブルはディスクユニットに添付されています。

標準フロッピィディスクユニット
本体背面
インターフェースボード

拡張ミニフロッピィディスクユニット(PC-80S32)を接続する場合は，PC-8801MKⅡ本体背面の拡張ミニフロッピィディスク用コネクタとPC-80S32を専用ケーブル(PC-80S32に添付)で接続します。

拡張ミニフロッピィ
ディスクユニット
本体背面

(注　意)

PC-80S32が増設できるのは，本体にミニフロッピィディスクが2ドライブ実装されているときに限ります。

(5) カセットテープレコーダ

PC-8801MKⅡとカセットテープレコーダを接続する場合は，PC-8801MKⅡに添付されている接続コード(PC-8093)を使用します。

接続するテープレコーダは，一般用オーディオカセットテープレコーダです。テープレコーダとして，パーソナルコンピュータ用に設計されたデータレコーダ（PC-DR311など）が操作しやすく失敗も少ないようですが，家庭用のものでも十分機能が果たせます。

本体背面

カセットテープレコーダ

赤

黒

白

カセットテープレコーダ背面

黒……リモート端子
白……イヤホーン端子
赤……マイク端子

4.4 拡張用スロット

PC-8801MKⅡには，拡張用スロットとして，本体後部に3スロット内蔵しています。この拡張用スロットには，標準フロッピィディスク用インタフェースボードや音声認識ボードなどを実装することができます。また，個人で設計した回路を接続することもできますが，その場合は，PC-8012-01ユニバーサルボードを使用してください。それ以外のボードですと，サイズやピン間隔が合わない場合などは，故障の原因となりますから注意してください。

注　意

スロット＃1，＃2とスロット＃3は信号が一部異っています。標準フロッピィディスク用インタフェースボードは，＃3に実装してください。拡張RAMボードなどは，どのスロットでも使用できます。

PC-8801MKⅡの拡張バススロットは，PC-8801と全く同じもので，また，PC-8012のマザーバスとは上位互換性を持っています。主な違いは，DMA関係の信号が加っている点と，PC-8012のINT0, INT1, INT2, INT6, INT7などが，割込拡張用信号やフロッピィディスク用の割り込み信号などに置き替えられている点などです。

PC-8801とくらべて

PC-8801の拡張用スロットは4つありましたが，PC-8801MKⅡでは，スロット＃4がなくなり，3つになっています。

(1) 拡張ボードの実装方法

PC-8801MKⅡの後部にはスロット・バスが3スロット有ります。ここには各種拡張ボードが入りますが，どのスロットも実装方法は同じです。(但しスロット1，2とスロット3では信号が異なります)。

① 作業を行う前にコンセントから，PC-8801MKⅡの電源プラグを抜いて下さい。

② スロット・バスのフタを外します。

・フタを止めている2本のビスを外すとフタは簡単に外れます。

・ドライバは大きさの合ったものを使用して下さい。大きさが合わないものを使用すると，ネジ山をつぶす原因となります。

③　拡張ボードを差し込みます。

・部品面(ICなどの部品が付いている面)を上にして，カード・ガイドのミゾにボードを合わせて差し込んで下さい。

・ボードが本体内部にほぼ隠れるところまでは軽く挿入できます。

・最後にカツンとショックがあるまで強く押し込みます。

・カードプラは押さないで下さい。破損の原因となります。

・ボードを軽く引っ張ってみて，抜けないことを確認して下さい。

④　スロット・バスのフタを閉めます。

・実装したボードの上から②で外したフタを閉め，ビスで止めて下さい。

(2)　拡張ボードの取り外し方

①　実装のときと同じ手順でフタを外して下さい。

②　拡張ボードにはカード・プラが付いていますので，これを図の矢印の方向へ動かすとボードは容易に外れます。(一部カード・プラが付いていない拡張ボードもあります)

カード・プラ

③　フタをして，ビスを止めてください。

第5章

第5章
システムの起動

この章では，PC-8801MKⅡのモードと，BASICを中心としたシステムの起動方法について説明します。

5．1　モード各種

5．1．1　モード

PC-8801MKⅡは，使用目的に応じていくつかのモードを選択して使うことができます。

(1)　BASICモード

N-BASIC，N_{88}-BASICの2つのBASIC言語を動作させるモードです。

(2)　ターミナルモード

PC-8801MKⅡをターミナルとして使うモードです。N_{88}-BASICから起動するターミナルモードと，N-BASICから起動するターミナルモードがあります。詳しくは，「第14章ターミナルモード」を参照してください。

(3)　機械語モニタモード

機械語データを直接取り扱うモードです。N-BASICまたはN_{88}-BASICから起動します。詳しくは，「第13章機械語モニタ」を参照してください。

5.1.2 BASICモード

PC-8801MKⅡは，N_{88}-BASICとN-BASICという2つのBASIC言語が使えます。これは，フロッピィディスクを使うことにより，N_{88}-BASIC，N-BASICのディスク版であるN_{88}DISK-BASIC，DISK-BASICに拡張できます。

N_{88}-BASIC, N_{88}DISK-BASICは，PC-8801のN_{88}-BASIC, N_{88}DISK-BASICと互換性がありPC-8801用に開発されたソフトウェアのほとんどを使用することが可能です。

また，N-BASIC，DISK-BASICは，姉妹機のPC-8001，PC-8001MKⅡ（N-BASICモード），PC-8801（N-BASICモード）に搭載されているものと同じで，現在数多く流通しているPC-8001用のソフトウェアを利用していただくために用意されています。

DISK-BASIC
N-BASIC
データの互換性あり
N_{88}DISK-BASIC
N_{88}-BASIC
拡張命令パッケージ
互換性のあるモード
PC-8001
PC-8001MKII（N-BASICモード）
PC-8801（N-BASICモード）
互換性のあるモード
PC-8801（N_{88}-BASICモード）

PC-8801MKⅡのBASICモードは，PC-8801のBASICモードと同じですが，新しい機能を追加するユーティリティとして，拡張命令パッケージが用意されています。これは，N_{88}-BASIC，N_{88}DISK-BASICにサウンド機能やタートルグラフィックス機能などを追加するもので，フロッピィディスクおよびカセットテープで供給されます。詳しくは，「第15章拡張命令」を参照してください。

5.2 BASICモードの起動

5.2.1 各BASICモードの起動方法

システム起動時に選択できるBASICモードは，次の表のとおりです。システム起動とは，電源を入れるか，リセット(リセットスイッチを押す)することです。各モードの起動には，ディップスイッチの設定や，システムディスクが必要なものがあります。

モード	ディップスイッチ (SW1)	ディップスイッチ (SW2)	システムディスク
N88DISK-BASIC	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	N88DISK-BASICシステムディスク
N88-BASIC		1 2 3 4 5 6 7 8	不要
DISK-BASIC	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	DISK-BASICシステムディスク
N-BASIC		1 2 3 4 5 6 7 8	不要

システム起動後にモードを切り替える方法もあります。

N88DISK-BASIC / N88-BASIC ＋ 拡張命令パッケージ ⇄(①②) N88DISK-BASIC / N88-BASIC →(③) DISK-BASIC / N-BASIC

① N88(DISK)-BASICに拡張命令パッケージを付加します。

◦フロッピィディスク使用の場合

拡張命令パッケージの入っているフロッピィディスクをドライブ番号1のフロッピィディスクドライブにセットします。

"@load. n88"をロードします。

load "@load. n88" ↵

プログラムを実行します。

run [⏎]

。カセットテープ使用の場合

本体添付(model 10)のカセットテープから，プログラムをロードします。

load "cas2：@exst" [⏎]

プログラムを実行します。

run [⏎]

これにより拡張命令パッケージが本体上にロードされ拡張命令が使えるようになります。

(参　考)

拡張命令の使用に関しましては本マニュアルの第15章「拡張命令」及びPC-8801MKII BASICリファレンスマニュアル第4章「拡張命令」の項を参照してください。

② 拡張命令パッケージを切り離します。

N88拡張(DISK)-BASICで次のように入力します。

cmd cut [⏎]

③ N88(DISK)-BASICから，DISK-BASICまたはN-BASICを起動します。

・DISK-BASIC

DISK-BASICのシステムディスクをミニフロッピィディスクのドライブ1にセットし，N88(DISK)-BASICで次のように入力します。

new on 1 [⏎]

・N-BASIC

N88(DISK)-BASICで次のように入力します。

new on 1 [ESC]+[⏎]

([ESC]キーを押しながら，[⏎]キーを押す)

ただし，[ESC]キーはしばらく押し続けていてください。

(注　意)

DISK-BASIC/N-BASICから，N88(DISK)-BASICへ切り替えることはできません。

5.2.2　N88DISK-BASICの起動方法

N88DISK-BASICの起動は，次の手順で行います。

(1) 「第4章本体と周辺装置との接続」に従って，本体のディップスイッチの設定および周辺装置との接続を確認します。(特に，N88-BASICモードにしておくことを忘れないでください。)

(2) PC-8801MKⅡの電源を「ON」にします。

(3) N88DISK-BASICのシステムディスクをセットしリセットスイッチを押します。

① ミニフロッピィディスクだけを使うとき

標準フロッピィディスクが接続されていれば，その電源を「OFF」にします。システムディスクは，必ずドライブ1にセットしてください。

② 標準フロッピィディスクも使うとき

通常は，標準フロッピィディスクユニットのドライブ1にセットしますが，それのドライブ2，3，4または，ミニフロッピィディスクのドライブ1でもかまいません。

(4) 数秒たつと，システムディスクをセットしたドライブのアクセス表示用LEDが点灯します。これは，ディスクコードの読み込みが始まったことを示しています。もし，読み込みが始まらない場合は，リセットスイッチを押してみてください。

(5) さらに数秒たつと，LEDが消え，同時にディスプレイにメッセージが表れます。

```
Disk version〔Aug 19, 1983〕
How many files(0-15)?
```

(このメッセージは，システムディスクによって多少異なる場合があります。)

(6) オープンするファイルの数を入力します。(詳しくは，「第10章入出力装置」を参照してください。)通常は⏎キーのみを入力してください。

(7) すると画面は次のようになり，N88DISK-BASICが使えるようになります。(表示される数値は異なる場合があります。)

```
Disk version〔Aug 19, 1983〕
How many files(0-15)?
NEC N-88 BASIC Version 1. 3
Copyright (C) 1981 by Microsoft
45382 Bytes free
OK.
```

(注　意)

N88DISK-BASICのシステムディスクは，model 20, model 30に添付されています。model 10で，ミニフロッピィディスクユニットを増設して使う場合は，別売のシステムディスク「PC-8837-2W」をお買い求めください。

(注　意)

(2)の段階で，本体前面のBASIC表示LEDが点灯していない場合は，N88DISK-BASICは起動できません。これは，DISK-BASICモードになっているためですので，ディップスイッチ1を設定しなおしてください。

5.2.3 DISK-BASICの起動方法

DISK-BASICの起動は，次の手順で行います。

(1) 「第4章本体と周辺装置との接続」に従って，本体のディップスイッチの設定および周辺装置との接続を確認します。(特に，N-BASICモードにしておくことを忘れないでください。)

(2) PC-8801MKⅡの電源を「ON」にします。

(3) DISK-BASICのシステムディスクを，ミニフロッピィディスクのドライブ1にセットしリセットスイッチを押します。

(4) 以後は，N88DISK-BASICをスタートさせるときと同じ要領です。

(注　意)

標準フロッピィディスクユニット用のDISK-BASICのシステムディスクはありません。また，DISK-BASICでは，標準フロッピィディスクユニットを使用することはできません。

(注　意)

DISK-BASICのシステムディスクは添付されておりません。

別売のDISK-BASICシステムディスク「PC-8034-2W」をお買い求めになるか，N88DISK-BASICシステムディスクの中に入っているユーティリティを使ってDISK-BASICシステムディスクを作ってください。

5.2.4 システムディスクのバックアップ

N88DISK-BASICをはじめて起動したときは，必ずシステムディスクのコピーをとる必要があります。システムディスクは1枚しか添付されていませんので，誤った操作でシステムディスクをこわしてしまうと以後使えなくなってしまいます。

フロッピィディスクのコピーを作ることをバックアップといいます。

バックアップはシステムディスクに入っているユーティリティプログラムを使って行います。

5.2.5　フロッピィディスクの初期化（フォーマット）

新しいフロッピィディスクをN88DISK-BASICまたはDISK-BASICで使うためには，まず初期化(フォーマット)をする必要があります。フォーマットを行うには，システムディスクに入っているユーティリティプログラムを使います。

(参　照)

ユーティリティプログラムの使い方は，本マニュアルの「付録3ディスクユーティリティの使い方」を参照してください。

5.2.6　N88-BASICの起動方法

(1) 本体のディップスイッチをN88-BASICモード，内蔵ミニフロッピィディスクドライブなしに設定します。
(2) さらに，標準フロッピィディスクユニットが接続されている場合は，標準フロッピィディスクユニットの電源を「OFF」にしておきます。
(3) PC-8801MKⅡの電源を「ON」にします。
(4) 以後は，N88DISK-BASICの起動方法と同じです。

5.2.7　N-BASICの起動方法

(1) 本体のディップスイッチをN-BASICモード，内蔵ミニフロッピィディスクドライブなしに設定します。
(2) PC-8801MKⅡの電源を「ON」にします。
(3) 次の画面表示になり，N-BASICが起動されます。

```
NEC PC-8001 BASIC Ver 1. 4
Copyright 1979 (C) by Microsoft.
OK.
```

5.3　機械語モニタの起動

機械語モニタはBASICを起動した後，コマンドによってモードの変更を行います。

N88DISK-BASIC
N88-BASIC
（＋　拡張命令パッケージ）
①→
←②
機械語モニタモード（N88版）

DISK-BASIC
N-BASIC
③→
←④
機械語モニタモード（N版）

①　N88(DISK)-BASICから機械語モニタモード(N88版)に入ります。

N88(DISK)-BASICで，次のように入力します。

mon ⏎

②　機械語モニタモード(N88版)からN88(DISK)-BASICへ戻ります。

コマンド待ちの状態で，次のように入力します。

[CTRL]＋[B]([CTRL]キーを押しながら，[B]キーを押す)

③　DISK-BASIC/N-BASICから，機械語モニタモード(N版)に入ります。

DISK-BASIC/N-BASICで，次のように入力します。

mon ⏎

④　機械語モニタモード(N版)から，DISK-BASIC/N-BASICへ戻ります。

コマンド待ちの状態で，次のように入力します。

[CTRL]＋[B]([CTRL]キーを押しながら，[B]キーを押す)

（参　考）

機械語モニタの使用方法に関しましては本マニュアルの第13章「機械語モニタ」の項を参照してください。

5.4　ターミナルモードの起動

ターミナルモードを起動するにはBASICモードからコマンドによってモードの変更を行う方法とディップスイッチの設定によってシステム起動時に選択する方法があります。

モード	ディップスイッチ（SW1）	ディップスイッチ（SW2）	システムディスク
ターミナル（N88ディスク版）	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	N88DISK-BASICシステムディスク
ターミナル（N88版）		1 2 3 4 5 6 7 8	不要

また，システム起動後にモードを切り替える方法もあります。

N88DISK-BASIC N88-BASIC （＋ 拡張命令パッケージ）	①→ ←②	ターミナルモード（N88版）
DISK-BASIC N-BASIC	③→ ←④	ターミナルモード（N版）

① N88（DISK）-BASICからターミナルモード（N88版）に入ります。

N88（DISK）-BASICで，term文を実行します。

② ターミナルモード（N88版）から，N88（DISK）-BASICへ戻ります。

ターミナルモードで次のように入力します。

SHIFT ＋ STOP （SHIFT キーを押しながら，STOP キーを押す）

③ DISK-BASIC/N-BASICから，ターミナルモード（N版）に入ります。

DISK-BASIC/N-BASICで，term文を実行します。

④ ターミナルモード（N版）から，DISK-BASIC/N-BASIC へ戻りま

す。

ターミナルモード(N版)で，次のように入力します。

GRPH＋B(GRPHキーを押しながら，Bキーを押す)

(参　考)

ターミナルモードの使用方法に関しましては本マニュアルの第14章「ターミナルモード」の項を参照してください。

5.5　電源のON/OFF

5.5.1　電源を入れる

電源を投入する手順は次のとおりです。

(1)　PC-8801MKⅡと各周辺装置を「第4章本体と周辺装置との接続」に従って接続します。

(2)　まず，プリンタを除く周辺装置の電源から「ON」にしていきます。

(3)　各周辺装置の電源を入れ終ったら，PC-8801MKⅡの電源を「ON」にします。

(4)　必要に応じてフロッピィディスクドライブにシステムディスクをセットします。セットが終ったらリセットスイッチを押して下さい。

(5)　もし，プリンタが接続されていれば，プリンタの電源を「ON」にします。

(注　意)

プリンタの電源は，最後に入れます。ただし，日本語ワードプロセッサなどでは，順序が異なるものがありますので，それぞれのマニュアルに従ってください。

Note **「電源の投入」は，「周辺装置」から始めて，「PC-8801MKII」の順序で行います。(ただし，プリンタを除く)**

もし，各装置の電源が入っているにもかかわらず，画面に何も表示されない場合は，すぐに，全装置の電源を「OFF」にして，もう一度，「第4章本体と周辺装置との接続」に従って，接続の仕方をチェックしてくださ

い。正しく接続されているにもかかわらず，ディスプレイに何も表示されない場合は，ディスプレイの「明るさ(BRIGHT)調整ツマミ」を回してみてください。

画面に何かが表示されてはいるが，読み取れない場合は，ディスプレイの電源の「ON」，「OFF」を行ったり，「水平同期調整ツマミ」を回してみてください。これでも画面に文字が表示されない場合は，ディスプレイの種類によるジャンパスイッチの選択がまちがっている可能性があります。全装置の電源を「OFF」にしてから，「4.3周辺装置との接続」の「(2)ディスプレイ」の項を参照して，ジャンパスイッチを調べてみてください。

5.5.2 電源を切る

電源を切ると，ユーザの入力したプログラムやデータはすべて消えてしまいます。電源を切るときは，以下のことを守ってください。

1. 重要なプログラムやデータを，カセットテープやフロッピィディスクにセーブし忘れていないか確かめます。
2. カセットテープやフロッピィディスクの読み書きの途中でないことを確かめます。読み書きの途中で電源を切ると，ファイルがこわれてしまいます。

（注　意）

フロッピィディスクユニットを使っている場合の特別な注意

N88DISK-BASICでは，オープンされたままのファイルがないことを確認してください。（念のためもう一度CLOSEを実行してください)。DISK-BASIC（N-BASICモードのディスクバージョン）では，REMOVEしたことを確認してください。

以上を確認したら，ドライブアクセス表示用LEDが消えていることを確かめたあと，フロッピィディスクをとり出してください。

これらをすべて確かめてから電源を「OFF」にします。電源を切る順序は，PC-8801MKⅡと周辺装置のどちらが早くてもかまいません。

電源を切った後，再度電源を入れる時は，5秒以上たってからにしてください。

5.5.3　タイマ用バックアップ電池

PC-8801MKIIのタイマ用バックアップ電池は，PC-8801MKIIの電源を連続して40時間以上「ON」にしておくと，完全に充電されます。また，完全に充電された後もPC-8801MKIIの電源を「ON」にしていても，バックアップ電池の機能を損なうことはありません。完全に充電されたバックアップ電池は，PC-8801MKIIの電源をずっと「OFF」にしたままでも，約2ヶ月の間，タイマを動かし続けます。ですから，1日に約40分程度PC-8801MKIIの電源を「ON」にするだけで，PC-8801MKIIのタイマは時を刻み続けます。

（参　照）

タイマの設定の方法は，「BASIC REFERENCE MANUAL」のDATE$, TIME$の項を参照してください。

（注　意）

タイマでバックアップしているのは，月，日，時，分，秒だけです。年は，システムディスクのユーティリティを使って初期設定してください。

5.6　リセットとウォームスタート

5.6.1　リセット

PC-8801MKⅡの本体前面の扉を開けると，リセットスイッチがあります。このスイッチを押すと，PC-8801MKⅡは電源を「ON」にしたときと同じ状態にセットされます。

このリセットスイッチは次のような場合に限り使用します。

1. 電源を「ON」にした直後。

   PC-8801MKⅡをはじめ，接続されている各周辺装置を初期状態にセットします。

2. キーボードから何も入力できなくなったとき。

   STOP キーをはじめ，どのキーを入力しても画面に何も現われないとき。

3. フロッピィディスクユニットにシステムディスクを読みに行かせるとき。

リセットスイッチを押すと，入力されていたプログラムやデータはすべて消えてしまいます。ですから，リセットスイッチの使用には十分注意してください。

Note **リセットボタンを押すと入力されていたプログラムやデータはすべて消えてしまいます。**

5.6.2 ウォームスタート

BASICモードで，STOPキーを押しながら，リセットスイッチを押すと，画面がクリアされ，左上スミに「OK」と表示されます。この機能を「ウォームスタート」といいます。ウォームスタートは,リセットスイッチを押したときとほぼ同じ機能を果しますが，入力されたプログラムやデータは保存されます。

ただし，ウォームスタートですと，フロッピィディスクユニットにシステムディスクを読みに行きませんから注意してください。

注　意

DISK-BASIC モードでウォームスタートを行うとフロッピィディスクへのアクセスができなくなりますから注意してください。

また，N88(DISK)-BASICからnew on文で, DISK-BASIC/N-BASICに切り替えた場合は，ウォームスタートさせることはできません。

注　意

ウォームスタートを行うと機械語を使ったプログラムの中には，動作しなくなるものがあります。

第6章

第6章
キーボード

6.1　キーの配置

PC-8801MKⅡのキーボードは，JIS配列に準拠しています。リターンキー(⏎)やシフトキー(SHIFT)など，良く使われるキーは押しやすい位置に，他のキーより大きなキーで配置されています。また，N88-BASICを使用しているときは，ヘルプキー(HELP)やコピーキー(COPY)など特別な機能を持ったキーが使用できます。さらに，PC-8801MKⅡのキーボードには，リピート機能が備わっていますから，同じキーを約1秒間以上押し続けると同じ文字が連続して表示されていきます。さらにキーボードの右側に数値入力用キー(テンキー)が備わっていますから，多くの数値を入力する場合に非常に便利です。

PC-8801MKⅡのキー配列を示します。

キー配列

6.2　キー入力

PC-8801MKⅡのキー入力は，ノーマルモード，グラフィックモード，カナモードの3つのモードがあります。

(1)　ノーマルモード

キーボードの通常の状態での入力で，シフトキー(SHIFT)を押したシフトポジションと合わせて，アルファベットや数字，特殊記号(＋，＊，？)などASCII標準キャラクタを入力するモードです。このとき，カナ キーはロックされていない(押されたままの状態になってない)ことに注意してください。

アルファベットは，そのまま押すと小文字が入力され，シフトキーを押しながら入力すると大文字が入力されます。CAPS キーを1度押すと，キーはロックされ，シフトキーを押さない状態でアルファベットの大文字が入力され，シフトキーを押しながら入力すると小文字が入力されます。この CAPS キーは，BASICのプログラムなどで，大文字を続けて入力する場合に使用すると大変便利です。CAPS キーのロックを解除する場合は，もう一度，CAPS キーを押します。

例

CAPS をロックしていない時

A チ →a

SHIFT ＋ A チ →A

CAPS をロックしている時

A チ →A

SHIFT ＋ A チ →a

(2)　グラフィックシンボル入力モード

GRPH キーを押しながら，キー入力を行うモードです。入力できるグラフィックシンボルは，「グラフィックシンボルのキー配列」を参照して下さい。グラフィックシンボル入力モードは，カナ キーや CAPS キーがロックされている状態でも動作します。

例

GRPH ＋ O ラ →♥

グラフィックシンボルのキー配列

(3)　カナ入力モード

[カナ] キーを押してロックした状態でキー入力を行います。また，ッ，ァ，ゥ，ェ，ォなどの右肩にあるカナ文字やカナ記号を入力したい場合は，[SHIFT] キーを押しながら入力します。(詳しくは,「6.3特殊キーの説明」を参照してください。)

例

[カナ] ＋ [A チ] → チ

[カナ] ＋ [SHIFT] ＋ [3 アァ] → ァ

注　意

複数のキーを同時に押した場合に，異なる文字が出ることがあります。

6.3 特殊キーの説明

SHIFT（シフトキー）

タイプライタのシフトキーと同じで，各キーの シフトポジションにある文字を入力する場合に使用します。また，アル ファベットを入力する場合は，大文字の入力に使用し，**CAPS** キーがロックされている状態では，逆に小文字の入力に使用します。

CTRL（コントロールキー）

他のキーと組み合わせて，特殊なコードを入力します。

キー入力	動　　　作
CTRL ＋A	HELP と同じ機能を果します。
CTRL ＋B	カーソルを1項目ごと左へ移動させます。
CTRL ＋C	プログラムの実行を中断してBASICのコマンドに戻ります。（ただし，入力待ちのときだけ有効です。通常は STOP キーを使用します。）
CTRL ＋D	現在，カーソルが点滅している項目を消去します。
CTRL ＋E	現在，カーソルが点滅している位置からその行の終りまでを消去します。
CTRL ＋F	カーソルを次の項目の先頭に移します。
CTRL ＋G	ブザー（ビープ音）を鳴らします。
CTRL ＋H	カーソルの前の1文字を消去します。 INS DEL と同じ機能です。
CTRL ＋I	タブ位置までカーソルを移動させます。タブ位置は8文字ごとに設定されています。
CTRL ＋J	ラインフィード（LFコード）を挿入します。 インサートモードでは，行の分割機能があります。（「第7章 スクリーン・エディタ」参照）
CTRL ＋K	カーソルをホームポジション（左上スミ）に移動させます。SHIFT ＋ HOME CLR と同じです。
CTRL ＋L	テキスト画面をクリアします。HOME CLR と同じです。
CTRL ＋M	リターンキー（↵）と同じです。
CTRL ＋O	テキスト画面の表示を中止します。再び CTRL ＋O を入力すると再開されます。（中止してから再開されるまでの表示されるデータは失われます。）

キー入力	動　　　作
CTRL＋R	インサートモードになります。SHIFT＋INS DELと同じです。
CTRL＋S	プログラムの実行の一時停止およびlist文，list文の実行の一時停止。STOP，COPY，SHIFT，カナ，GRPH，CTRL＋C以外のキーを入力すれば実行が再開されます。
CTRL＋U	カーソルが点滅している行を一行全部消去します。
CTRL＋X	カーソルが点滅している行の最後の文字の次にカーソルを移動させます。

これらのコントロール文字は，N88BASICを使用しているときにこのように動作します。N-BASICを使用しているときは多少異なります。

○異なった動作を示すもの

CTRL＋R ス　カーソルの前へスペースを1個挿入します。

CTRL＋J マ　行を2つに分けます。

○動作しないもの

CTRL＋A チ，CTRL＋D シ，CTRL＋F ハ，

CTRL＋O ラ，CTRL＋U ナ，CTRL＋X サ

○N-BASICだけで動作するもの

CTRL＋N ミ　カーソルを次の項目の先頭へ移します。

N88BASICのCTRL＋F ハと同じです。

⏎リターンキー(RETURN)

このキーの入力により，1行の入力が終ります。このキーを入力するとカーソルは，次の行の先頭に移動します。

カナ カナキー

このキーを押すと，キーはロックされ，入力はカナモードになります。キーに表示されている文字の位置により，SHIFT キーとカナキーの組み合わせを表しています。

ロックされたカナキーを押すともう一度押すと，カナキーのロックは解除されます。

例

3 ア → 3

SHIFT + 3#アァ →#

カナ + 3#アァ →ア

カナ + SHIFT + 3#アァ →ァ

Aチ →a

CAPS + Aチ →A

SHIFT + Aチ →A

CAPS + SHIFT + Aチ →a

カナ + Aチ →チ

STOP ストップキー

プログラムの実行を停止するときに使用します。

HOME CLR ホーム・クリアキー

このキーを入力するとテキスト画面がクリアされ，カーソルは画面の左上スミに移ります。SHIFT を押しながら入力すると，テキスト画面はクリアされずカーソルだけ移動します。

→ ← ↑ ↓ カーソル移動キー

→を入力すると右へカーソルを移動します。

←を入力すると左へカーソルを移動します。

↑を入力すると上へカーソルを移動します。

↓を入力すると下へカーソルを移動します。

SHIFT + →を入力すると←と同じ動作をします。

SHIFT + ↑を入力すると↓と同じ動作をします。

画面の右端にカーソルがあるとき，→を入力するとカーソルは次の行の左端に移動します。画面の右下スミにあるときに，→を入力すると画面は1行上へスクロールし，カーソルは最下位の行の左端に移動します。

画面の左端にカーソルがあるとき，←を入力するとカーソルは1つ前の行の右端に移ります。最上位行の左端にあるときは，移動しません。

画面の最上位行にカーソルがあるとき，↑を入力してもカーソルは移動しません。画面の最下位行にカーソルがあるとき，↓を入力した場合も同じです。

INS/DEL インサート/デリートキー

このキーを入力すると，点滅しているカーソルの左の文字が1文字消され，カーソルより右の1行分の文字列を左につめていきます。

SHIFT ＋ INS/DEL を入力すると，インサートモードになり，カーソルが点滅している文字とその前の文字との間に入力された文字を挿入します。もう一度 SHIFT ＋ INS/DEL を入力するか，カーソル移動キーもしくは，リターンキーを入力するとインサートモードから抜け出します。

（注　意）

N-BASICモードでは，SHIFT ＋ INS/DEL で，カーソルの前にスペースを1個挿入します。

連続した多くの文字を消去する場合に，INS/DEL を押し続けますと，キーを離した後も文字を消し続ける場合があります。このような場合は，CTRL ＋ Hｸ をお使いください。CTRL ＋ Hｸ は INS/DEL と同じ動作をし，キーを離すと同時にその動作をやめます。

CAPS キャピタルロックキー

このキーを押してロックすると，以後，アルファベットを入力すると大文字が入力されます。このとき SHIFT を押しながら入力すると小文字が入力されます。CAPS のロックを解除する場合はもう一度このキーを押します。

f･1 ～ f･5 ファンクションキー

SHIFT キーといっしょに使用する場合を含めて，10種類の入力を行うファンクションキーとして使用できます。

このファンクションキーを押すと，定義されている文字列が入力されます。ですから，よく使う文字列をセットしておけば，プログラムやデータなどを入力する時間がずっと短くなります。各ファンクションキーは，最大15文字まで定義できますが，定義されている内容は，40キャラクタモードのときで5文字，80キャラクタモードでは，11文字まで表示されます。

ファンクションキーを新しく定義する場合は，次のように入力します。

key △ 4,"run"＋chr$(13) ⏎

（△は，スペースを1つ開けることを表しています。）

これで，f・4に“runC_R”が定義されましたね。（key文およびchr$については，「BASIC REFERENCE MANUAL」を参照してください。）

また，現在，定義されている各ファンクションキーの内容を表示させるには，次の命令を実行します。

key △ list ⏎

N88-BASICおよびN-BASICモードにおいてファンクションキーは，次のようにセットされています。

	N88-BASIC	N-BASIC
f・1	load"	H_T
f・2	auto	auto
f・3	go to	go to
f・4	list	list
f・5	runC_R	run C_R
f・6	save"	time $
f・7	key	key
f・8	print	print
f・9	edit. C_R	list. ↑ ↑
f・10	cont C_R	cont C_R

f・6～**f・10**は **SHIFT** キーを押した場合です。N-BASICの **f・1**「H_T」は，**TAB** と同じ機能です。

GRPH グラフキー

グラフィックシンボルを入力するときに使います。

ESC エスケープキー（N88-BASICモード）

エスケープコードを入力するときに使用します。

（注　意）

N-BASICモードでは，プログラムの実行もしくは処理を一時停止させるときに使用します。再開は，任意のキーを入力します。

ROLL UP ロールアップキー（N88-BASIC BASICモード）

ROLL DOWN ロールダウンキー(N88-BASICモード)

N88DISK-BASICではEDIT文とともに用いて，スクリーンエディタによるプログラムの修正に使用します。詳しくは，「第7章スクリーン・エディタ」を参照してください。

また，モニタのEコマンドでも，同様の用途に使用します。これ以外の場合は，テキスト画面を上下にスクロールさせるだけです。

TAB タブキー(水平タブキー)(N88-BASICモード)

このキーを入力すると，カーソルは8文字単位で右へ移動します。

COPY コピーキー(N88-BASICモード)

ディスプレイの画面をプリンタへコピーします。この場合,テキスト画面とグラフィック画面の両方をプリントします。(N88-BASICの“copy 3”という命令と同じです。)

テキスト画面だけをコピーする場合は，

CTRL＋COPY(N88-BASICの“copy 1”と同じです。)

と入力します。グラフィック画面だけの場合は，

GRPH＋COPY(N88-BASICの“copy 2”と同じです。)

と入力します。

(注 意)

COPY(またはcopy文)でプリンタへ出力している途中で，STOPまたはCTRL＋C で出力を中止した場合は，プリンタの電源の「OFF」「ON」を行ってプリンタを初期化してください。そうでないと，以後，プリンタが正常に動作しない場合があります。

HELP ヘルプキー(N88-BASICモード)

ヘルプキーには，2つの機能があります。

1. エラー位置の検出

   プログラムの実行中にエラーが起きた場合に，その文と，場所を表示します。(詳しくは，「第7章スクリーンエディタ」を参照してください。)

2. ON HELP GOSUB文

   input文などで入力待ちになったときなどに利用する特別なキー

割込みです。(詳しくは，「BASIC REFERENCE MANUAL」を参照してください。)

第7章

第7章
スクリーン・エディタ

N88-BASICには，ディスプレイを活用して，プログラムの作成および修正が，簡単に行えるように，「スクリーン・エディタ」と呼ばれる画面編集機能を備えています。この機能により，修正したい行を改めて最初から入力する必要がなくなり，プログラムの入力や訂正も短時間で行えます。

7.1　入力中の修正

キーボードからプログラムなどを入力しているとき，誤って隣りのキーを押してしまったり，途中で文字を抜かしてしまう場合があります。こんな時，その行を最初から入力しなおしていては大変です。そこで，スクリーン・エディタの機能を使います。

(1)　カーソルの直前の文字の消去

文字を入力しているときに入力した文字を消去するには，[INS DEL]を押します。すると，カーソルの前の1文字が消去されます。[CTRL]+[H]でも同じ機能を実現できます。

(2)　途中の文字の修正

途中で押しまちがえた文字を正しい文字に置き替えたい場合は，カーソル移動キー([←][→][↑][↓])を使用して，まちがった文字のところへ移動させ，正しい文字に置き替えた後，再びカーソル移動キーで，入力を中断したところへ戻って入力を続けます。

(3)　途中の文字の消去

途中で押しまちがえた文字を消去したい場合は，カーソル移動キー([←][→][↑][↓])を使用して，消去したい文字の次の文字の所へ，カーソルを移動させて，[INS DEL]を入力します。この場合，カーソルの前の文字が消去され，カーソルから右の文字列が1文字分，左へ移動します。

(4) 文字の挿入

何文字か途中で挿入したい場合は，文字を挿入したい位置の次の文字にカーソルを移動させ，SHIFT + INS DEL を入力します。これでインサートモードとなり，これから後で入力される文字をカーソルの前に挿入していきます。インサートモードから抜け出すには，再び，SHIFT + INS DEL を入力するか，カーソル移動キーかリターンキー(⏎)を入力します。(SHIFT + INS DEL は，CTRL + R ス で代用できます。)

スクリーン・エディタの機能を用いてプログラムの訂正を行った場合は，各行番号の訂正が終る度に，必ずリターンキー(⏎)を入力してください。PC-8801MKⅡは，リターンキーの入力により，その文での変更を理解します。

Note 1つの行番号の文での，入力や訂正が終ったら，他の行番号の文に移る前に，必ずリターンキー(⏎)を入力してください。

例

「10 prannt Hello」を「10 print "Hello"」に訂正してみましょう。(□はカーソルを表しています。)

10 prannt Hello □

← でカーソルを移動させます。

10 pr[a]nnt Hello

I = を入力します。

10 pri[n]nt Hello.

→ でカーソルを移動させます。

10 prin[n]t Hello

INS DEL で，カーソルの前の"n"を消去します。

10 pri[n]t Hello

→ でカーソルを移動させます。

10 print [H]ello

SHIFT + INS DEL で，インサートモードにした後，SHIFT + 2 フ を入力します。

10 print "[H]ello

→ でカーソルを移動させます。これで，インサートモード

から抜け出しています。

10 print “Hello□

[SHIFT]＋[2 "] を入力します。これでOKですね。

10 print “Hello” □

他の行番号の文に移る前に，リターンキーを入力します。これで，訂正した後の文が，メモリに記憶されました。

（注　意）

プログラムを訂正している時に，カーソルの動かし方によっては，2つの行がつながってしまう場合があります。

特に，[→]キーを用いてスクリーンの右端を越えてカーソルを移動させたような場合には注意が必要です。

7.2　エラーの訂正

プログラムに何らかのエラーがある場合，その文で実行が中断されます。このとき，N88-BASICは，エラーの種類と，そのエラーが起きた行番号をディスプレイに表示します。さらに，エラーの種類によっては，その行を表示し，エラーのある項目か，その次の項目の先頭に，カーソルを表示します。

20 PRINT A：PLINT B：GOTO 10

この場合ですと，カーソルのすぐ前の項目，すなわち，“PLINT”の中にエラーがありますね。そうです，“R”が，“L”とまちがって入力されていますね。そこで，[←]でカーソルを移動させ，[R ス]を入力して，リターンキーを押して訂正を終ります。

エラーのあった行を表示しない場合は，[HELP]を入力します。すると，エラーメッセージで表示された文番号の文を表示して，エラーのある項目か，その次の項目にカーソルを表示します。

N88DISK-BASICを使用しているときには，プログラムの訂正に，より便利な「エディットモード」が準備されています。まず，訂正したいプログラムを入力またはロードします。（list [↵] 又は [f・4]＋[↵] で内容が表示されます。）ここでわかり易くするために一度，画面を消します。（[HOME CLR] 又は，CLS [↵]）

次にeditコマンドを実行します。

edit ⏎

すると，最後の行が画面の一番上に表示されます。次の行が見たい場合は，ROLL DOWN キーを入力します。このキーを押し続けると，どんどん前の行が表示されて行き，画面に入りきらない分は，どんどん下へ消えて行きます。しかし，エディットモードに入っている時は，ROLL UP キーを入力することにより，一度画面の下へ消えていった行をもう一度見ることができます。そして訂正したい行を見つけたら，その位置にカーソルを移動させて，INS DEL キーなどを使って誤りを直します。このときも，訂正が終ったら必ず⏎キーを入力してください。エディットモードから抜け出したい場合は，HOME CLR キーを入力します。また，とても長いプログラムなどで，途中からエディットモードをスタートさせたい場合は，

edit △ 〔行番号〕 ⏎

と入力します。すると，その行番号の文が表示されますから，以後，ROLL UP，ROLL DOWN キーを使って前後の行を表示させ，訂正します。

N_{88}DISK-BASICを使用していない場合は，ただ単に画面を上下にスクロールさせるだけですから，listコマンドなどで表示させたプログラムなども，ROLL UP，ROLL DOWN，キーで一度画面の外へ追い出してしまうと，ROLL UP，ROLL DOWN キーでは再び見ることはできません。

7.3　便利な編集機能

(1)　1行消去CTRL＋Uナ

カーソルが点滅している行を消去する場合は，CTRL＋Uナを入力します。これは，その行の先頭から，最後(リターンキーが入力された所)までを画面から消去します。この場合は，行番号も消えてしまいますからリターンキーを入力してもプログラムは訂正されません。

(2)　カーソルから，その行の終わりまでの消去CTRL＋Eイ

カーソルが点滅している文字から，その行の終わりまでを消去する場合は，CTRL＋Eイを入力します。この後で，リターンキーを入力すれば，メモリの内容も訂正されます。ですから行番号のすぐ次の位置にカーソルを移動し，CTRL＋Eイを入力し，その後リターンキーを入力するとその行番号の文はプログラムから削除されます。

(3)　カーソルの移動　CTRL＋Fハ，　CTRL＋Bコ，　CTRL＋Xサ

PC-8801MKⅡのキーボードには，オートリピート機能が備っているため，カーソル移動キーを押し続けると，カーソルはどんどん移動します。しかし，さらに便利な機能として，項目ごとの移動ができます。

CTRL＋Fハ を入力すると，カーソルは次の項の先頭へ移動し，CTRL＋Bコ を入力すると1つ前の項目の先頭へ移動します。また，CTRL＋Xサ を入力すると，その行の最後の文字の次にカーソルが移動します。

CTRL＋Fハ は，N-BASICの CTRL＋Nミ と同じ動作を行います。N_{88}-BASICでは，CTRL＋Nミ は動作しません。

(4)　行の分割とフィードの挿入CTRL＋Jマ

1行の内容を2行にわたるように書き直したい場合には，区切りたい箇所でカーソルを止め，SHIFT＋INS DEL でインサートモードに切り替え，CTRL＋Jマ を押すと，カーソルから後のその行の最後までの文が次へ移動します。移動した行の先頭にライン番号を付加し，⏎を押せば，これまでの1行の内容が2行にわたって書かれたことになります。

また，インサートモードになっていないときに，CTRL＋Jマ を入力すると，その行の最後にフィードを挿入します。

(5)　1項目の削除 CTRL ＋ D

カーソルの点滅している文字から，その項目の終わりまでを消去します。

7．4　N-BASICモードのスクリーン・エディタ

N-BASICモードでも，N88-BASICモードと同じようなスクリーン・エディタの機能を備えています。ただし，edit文はなく，ROLL UP，ROLL DOWN が動作しませんし，HELP の機能もありません。ですから，エラーメッセージで示された文を訂正する場合は，list文を使ってディスプレイに文を表示した後，カーソルを移動させて訂正を行います。詳しくは，「PC-8001ユーザーズマニュアル」か「N-BASIC REFERENCE MANUAL」を参照してください。

第8章

第8章
ディスプレイモード

N88-BASICは，いろいろなディスプレイモードを備えています。特に，グラフィック画面の表示用に独立したビデオRAM(V-RAM)を48Kバイト備えているので，複雑な表示も可能となり，また，テキスト画面と独立して扱えます。さらに，専用高解像ディスプレイを接続することにより，640×400ドットの鮮明な画像を表示することができます。

N88-BASICのディスプレイモードは次のように分類されます。

- テキスト画面
- グラフィック画面
  - グラフィックモード
  - カラーグラフィックモード
  - 専用高解像度ディスプレイモード

このなかで，専用高解像度ディスプレイモードは，専用高解像度ディスプレイを接続したときだけ選択できます。また，PC-8801MKⅡに，家庭テレビ用カラーアダプター(PC-8044K)を接続すれば，家庭用T. V. を使用できますが，その場合は，画面の設定によっては，文字等が見にくい場合があります。

8.1　テキスト画面

キーボードから入力した文字や，プログラムのリストなどを表示する画面を「テキスト画面」と呼びます。テキスト画面は，ユーザーズメモリに設定されたテキストVRAMを使用して表示しますから，グラフィック画面とは独立しています。ですから，テキスト画面だけを表示させることも，グラフィック画面にテキスト画面を重ねて表示することも可能です。(詳しくは，「8.2グラフィック画面」および「8.4モノクロディスプレイ用コネク

タへの出力の選択」を参照してください。)

8．1．1　テキスト画面の〈文字数/行〉,〈行数/画面〉の指定

テキスト画面に表示できる文字数は,次の4つの組み合わせがあります。

80文字×25行，80文字×20行(80キャラクタモード)

40文字×25行，40文字×20行(40キャラクタモード)

電源を「ON」にしたとき，またはリセットボタンを押したときの〈文字数/行〉は，本体前部のDIPスイッチで選択できます。

SW1
1 2 3 4 5 6 7 8
80文字/行

SW1
1 2 3 4 5 6 7 8
40文字/行

SW1
1 2 3 4 5 6 7 8
25行/画面

SW1
1 2 3 4 5 6 7 8
20行/画面

この〈文字数/行〉,〈行数/画面〉は，width文で変更します。

width △〈文字数/行〉〔,〈行数/画面〉〕⏎

(△は，1文字分のスペースを開けることを表しています。)

例えば，80文字×20行の画面を40文字×20行に変更する場合は，次のように入力します。

width △ 40, 20 ⏎

しかし，width文では，行数を変更しないパラメータは省略してもかまいませんから，次のように入力しても同じ動作をします。

width △ 40 ↵

テキスト画面の表示文字数の指定は，width文で行います。

width △ 〈文字数/行〉〔，〈行数/画面〉〕

（ただし，行数/画面を変更しない場合は省略できます。）

（注　意）

家庭用T. V.または家庭T. V.用ブラウン管を使用しているモニタを接続している場合は，40キャラクタモードを指定してください。80キャラクタモードでは，文字が不鮮明で見にくいことがあります。

8．1．2　テキスト画面のモード設定

テキスト画面のいろいろなモードをconsole文でセットします。電源を「ON」にしたときや，リセットボタンを押したときのテキスト画面のモードは次のようにセットされています。

スクロールのスタート行………0
スクロールする行数……………20
ファンクションキーの表示……1（表示する）
カラー／モノクロモード………0（モノクロモード）

この4つの値をconsole文で指定できます。

テキスト画面のモードの設定はconsole文で行います。

console △〔〈スクロールのスタート行〉〕
，〔〈スクロールする行数〉〕
〔，〈ファンクションキーの表示〉〕
〔，〈カラー/モノクロモード〉〕

まず，「スクロール」について説明します。console文で指定しない限り，最下位行まで文字を表示して改行を行ったり，最下位行の端まで文字を表示し，さらに文字を表示させようとした場合は，画面全体が1行上へ移動し，最上位の行が消えてしまいます。この動作を「スクロール」といいます。このままですと，どんどん文字を表示させていくと，画面に残しておきたいと思った文字も，どんどん上へスクロールされていき，いずれ消えてし

まいます。そこで,残しておきたい文字を画面に表示したあと,console文で,〈スクロールのスタート行〉と〈スクロールする行数〉を設定して,残しておきたい部分がスクロールしないようにします。行は,最上位から0,1,2……と数えます。

例

最上位行から10行を残しておきたい場合。

第0行から第9行の10行を残すように設定しますから,〈スクロールのスタート行〉は10,〈スクロールする行数〉は,20行/画面ならば,残り10行ですから,これも10となります。

console △ 10, 10 ↵

これで,上半分の10行は,スクロールされません。もとの状態にもどすには,第0行から20行スクロールするようにセットします。

console △ 0, 20 ↵

〈ファンクションキーの表示〉は,画面の最下位の行に,ファンクションキーの内容を表示させるかどうか設定します。表示する場合は1,表示させない場合は0を指定します。

例

ファンクションキーの表示を消す場合。

console △ , , 0 ↵

ファンクションキーを表示させる場合。

console △ , , 1 ↵

〈カラー/モノクロモード〉の切り替えは,color文の〈ファンクションコード〉の機能を規定します。「8.1.3テキスト画面のカラー指定」を参照してください。

例

カラーモードの指定をする場合。

console △ , , , 1 ↵

モノクロモードの指定をする場合。

console △ , , , 0 ↵

注　意

console文でもwidth文と同じように，値を変更しないパラメータは省略できます。ただし，変更するパラメータより前のコンマは省略できませんから注意してください。

例

5行めから15行めまでをクスロールさせ，ファンクションキーは表示させず，カラーモードの指定を行う場合。

console △ 4, 10, 0, 1 ⏎

8.1.3　テキスト画面のカラー指定

テキスト画面のカラー指定は，color文で行います。N88-BASICには，3種類のcolor文があります。このうち，テキスト画面のカラー指定を行うのは，次の2つです。

(i)　ファンクションコード/フォアグラウンドカラー/バックグラウンドカラーの指定を行うcolor文。

color △〔<ファンクションコード>〕〔, <バックグラウンドカラー>〕〔,〕〔, <フォアグラウンドカラー>〕

この中で，テキスト画面に関係があるのは，<ファンクションコード>だけで，他はグラフィック画面に関係あるものです。ですから，ここでは，<ファンクションコード> だけを説明し，残りのパラメータは，「第9章グラフィックス」の中で説明します。

「8.1.2テキスト画面のモード設定」で説明したconsole文を思い出してください。その4番めのパラメータに <カラー/モノクロモード> の指定がありましたね。この指定により，<ファンクションコード> の指定の意味が異なってきます。

(1)　console文で，カラーモードの指定がなされていると，<ファンクションコード> は <カラーコード> として，表示する文字の色を指定できます。

〈ファンクションコード〉	色
0	黒
1	青
2	赤
3	紫
4	緑
5	水色
6	黄色
7	白

(2) console文でモノクロモードを指定している場合は,〈ファンクションコード〉は，文字の表示モードを指定します。

〈ファンクションコード〉	文字の表示コード
0	ノーマル
1	シークレット
2	ブリンク
3	シークレット
4	リバース
5	リバース，シークレット
6	リバース，ブリンク
7	リバース，シークレット

〈ノーマル〉とは，通常の表示モードです。〈リバース〉にすると背景と文字が逆転します。〈シークレット〉は，入力された文字を表示せず，文字数分だけカーソルが移動していきます。〈ブリンク〉は文字を点滅させます。

例

テキスト画面の文字を赤にしたい場合。

console △ ,,, 1 [↵]　(カラーモード)
color △ 2 [↵]　(赤)

この2つの命令により，これ以後入力された文字は，赤で表示されます。

例

暗証番号のように，ディスプレイには表示させたくない場合。

console △ ,,, 0 [↵]　(モノクロモード)
color △ 1 [↵]　(シークレット)

電源を「ON」にしたとき，または，リセットボタンを押したときは，モノクロモードとなっていますから，このconsole文は省略できます。

Note **テキスト画面の文字のカラーの指定および表示モードの指定は，color文で行います。**

color △ 〈ファンクションコード〉⏎

(ii) テキスト画の区画を指定した〈ファンクションコード〉による指定。

N_{88}-BASICでは，テキスト画面のある区画の〈ファンクションコード〉を指定することができます。

color@ (X_1, Y_1)−(X_2, Y_2)，〈ファンクションコード〉⏎

(X_1，Y_1)，(X_2，Y_2)は，キャラクタ座標です。

〈ファンクションコード〉については，(i)で説明したとおりです。キャラクタ座標は，「何文字めで何行め」というように位置を指定するもので，画面の左上スミが，(0,0)となります。

例

80文字×20行モードのとき，画面の右上半分の文字を青にする場合。

(0，0)　(40，0)　(79，0)
(40，10)
(0，10)　(79，10)
(0，19)　(79，19)
(40，19)

80文字×20行モードのキャラクタ座標は上図のようになります。区画を指定するときは，その区間の左上スミと右下スミを指

定します。

console △ ，，，**1** ⏎

color @ （**40, 0**）−（**79, 10**），**1** ⏎

同じ区画に表示されている文字を点滅(ブリンク)させる場合は，次のようになります。

console △ ，，，**0** ⏎

color @ （**40, 0**）−（**79, 10**），**2** ⏎

区画を指定したテキスト画面の〈ファンクション コード〉による指定。

color @ （X_1, Y_1）−（X_2, Y_2），**〈ファンクションコード〉** ⏎

（X_1, Y_1）

（X_2, Y_2）

［座標はキャラクタ座標で指定します。
(0,0)〜(〈文字数/行〉 −1，〈行数/画面〉 −1)］

注　意

このcolor文で表示されている文字の色を変えても，次にこの区間に表示される文字は，(i)のcolor文で指定された〈ファンクションコード〉によって表示されます。

8.1.4　テキスト画面に関する注意

テキスト画面のカラー指定を行うとき，カラーモード，モノクロモードにかかわらず，1行内で，19ケ所以上，変化点を指定するとそれ以上同一の色になったり，同一の表示モードになってしまいます。ですから，多く使う色または表示モードを，区画を指定するcolor文で指定しておいた方が良いでしょう。

8．1．5　N-BASICのテキスト画面

N-BASICのディスプレイモードは，1つの画面に，テキストとグラフィックの両方を表示させるという点で，N$_{88}$-BASICとは全く異なっています。しかし，テキストの表示に関しては，ほとんどN$_{88}$-BASICと共通です。

width文とconsole文はN$_{88}$-BASICと全く同じです。ただし，N-BASICでは，width文で，80文字/行，40文字/行以外に，36文字/行，72文字/行を指定できます。

color文は1種類しかなく，次の形で入力されます。

color〈ファンクションコード〉,〈ヌルキャラクタ〉,〈グラフィックスイッチ〉 ⏎

〈ファンクションコード〉については，N$_{88}$-BASICと全く同じです。〈ヌルキャラクタ〉は，画面をクリアしたときに，画面を埋めるキャラクタを指定します。〈グラフィックスイッチ〉は，画面をグラフィック用画面として設定するか，テキスト表示用として設定するのかを指定します。

8.2　グラフィック画面

N$_{88}$-BASICは，より優れたビジネスグラフィックスを提供できるように，多彩なグラフィック機能を備えています。まず，N$_{88}$-BASICのグラフィック画面のモードには，〈グラフィックモード〉と〈カラーグラフィックモード〉があります。さらに，専用高解像度ディスプレイを使用している場合は，640×400ドットというきわめて鮮明な画像が得られる〈専用高解像度ディスプレイモード〉を選択することができます。

グラフィック画面のモード選択は，screen文を使います。

screen △〔〈**パラメータ1**〉〕〔,〈**パラメータ2**〉〕〔,〈**パラメータ3**〉〕
〔,〈**パラメータ4**〉〕 ↵

(i)　パラメータ1……モードの選択

パラメータ1	モード
0	カラーグラフィックモード
1	グラフィックモード
2	専用高解像度ディスプレイモード

(ii)　パラメータ2……グラフィック画面の表示の有無。高速書き込みモードの指定。
ビット指定となっています。

○第0ビット……高速書き込みモードの指定

第0ビット	高速書き込みモード
0	指定しない
1	指定する

○第1ビット……グラフィック画面の表示

第1ビット	グラフィック画面の表示
0	行う
1	行わない

前項の表ではわかりにくい方は，次の表を参考にしてください。

パラメータ2	機　　　　能
0	高速書き込みではないモードで，グラフィック画面を表示させせます
1	高速書き込みモードでグラフィック画面を表示させせます。
2	高速書き込みではないモードですが，グラフィック画面は表示させません。
3	高速書き込みモードですがグラフィック画面は表示させません。

参　照

「高速書き込みモード」については，「8.3高速書き込みモード」で説明します。

(iii)　パラメータ3……グラフィックモードでグラフィック命令を実行する画面を指定します。

パラメータ	書き込む画面
0	第1画面
1	第2画面
2	第3画面

(iv)　パラメータ4……グラフィックモードで表示する画面を指定します。パラメータの値と表示する画面は，ビット対応になっています。

ビット	0	1	2
画面	第1画面	第2画面	第3画面

各ビットが，1の場合に対応する画面を表示します。ですから，実際に入力する数値と表示される画面の関係は，次のようになります。

パラメータ4	表示する画面		
	第1画面	第2画面	第3画面
0			
1	○		
2		○	
3	○	○	
4			○
5	○		○
6		○	○
7	○	○	○

（参　照）

〈パラメータ3〉および〈パラメータ4〉は，〈パラメータ1〉で，グラフィックモードを選択したときに有効となります。詳しくは，「8.2.1グラフィックモード」を参照してください。

Note

screen命令を実行する前のグラフィック画面は，次のモードにセットされています。

screen △ 0, 0

8.2.1　グラフィックモード

N88-BASICのグラフィックモードは，640×200ドットの画面を3画面持っていて，テキスト画面のカラー指定を利用すると，キャラクタ単位(8×8ドット単位)で8色まで指定できます。3画面のうち，どの画面でグラフィック命令を実行し，どの画面をディスプレイに表示させておくかを〈パラメータ3〉と〈パラメータ4〉で指定します。

（例）

第1画面でグラフィック命令を実行させながら，その画面をディスプレイに表示させる場合。

screen△ 1, 0, 0, 1 ⏎

例

第3画面でグラフィック命令を実行させながら，第1画面と第2画面を表示させる場合。

screen△ 1, 0, 2, 3 ⏎

この場合は，実行されているグラフィック命令の結果はディスプレイには表れません。実行終了後に次の命令を実行すると，第3画面で実行された結果も合わせて表示されます。

screen△ 1, 0, 2, 7 ⏎

参　照

グラフィック命令については，「第9章グラフィックス」および「BASIC REFERENCE MANUAL」を参照してください。

グラフィック画面のカラー指定については，「8.1.3テキスト画面のカラー指定」を参照してください。

Note

グラフィックモードを指定するときは，使用する画面も指定します。

screen△ 1, 0, 〈パラメータ3〉，〈パラメータ4〉⏎

注　意

次の場合，SCREEN文のパラメータ4によって指定した表示画面の設定はリセットされます。(3画面が合成されて表示されます。)

- WIDTH文の実行
- テキスト画面のカラー/モノクロモードの切換え(CONSOLE文)
- CMD TEXT ON文で，テキスト画面を再表示させた場合(CMD SING "X1", CMD CUTも含む)。

このような場合には，各命令の実行直後，SCREEN1を実行してグラフィックモードの再設定を行ってください。

8.2.2　カラーグラフィックモード

N88-BASICのカラーグラフィックモードは，640×200ドット１画面で構成されています。カラーグラフィックモードでは，各ドット単位に８色のカラー指定ができますから，細かなグラフィック模様を鮮やかに表示でき，複雑なグラフも簡単に表示できます。また，paint文を使用すると，中間色やタイリングも簡単に表示できます。詳細は，「第９章グラフィックス」で説明してあります。

カラーグラフィックモードの指定は，次の命令を実行します。

screen△ 0, 0 ⏎

8.2.3　専用高解像度ディスプレイモード

PC-8801MKⅡに専用高解像度ディスプレイを接続した場合は，N88-BASIC使用時に，640×400ドットの〈専用高解像度ディスプレイモード〉を選択できます。

専用高解像度ディスプレイモードでは，一画面に40文字×20行もの漢字を表示できます。(グラフィックモード，およびカラーグラフィックモードでは，40文字×10行までしか表示できません。)また，テキスト画面のカラー指定を利用して，キャラクタ単位(８×８ドット単位)に８色まで指定できます。

専用高解像度ディスプレイモードを指定する場合は，次の命令を実行します。

screen 2, 0 ⏎

8.3　高速書き込みモード

ディスプレイに表示させながらグラフィック命令を実行させる場合，グラフィック用VRAMは，N88-BASICからのグラフィック命令の実行結果を記憶すると同時にディスプレイにもデータを渡しています。ですから，ディスプレイの画面を乱さないように動作させると，実行時間が少し長くなってしまいます。そこで，画面が多少乱れても，速度を優先させるのが，高

速書き込みモードです。このモードは，どのグラフィック画面のモードでも指定できます。

高速書き込みモードの指定は，screen文の〈パラメータ 2〉で奇数を指定します。

8.4　モノクロディスプレイ用コネクタへの出力の選択

N88-BASICは，モノクロディスプレイ用コネクタに出力する画面をハードウェアで設定できます。この設定は，本体前部のジャンパー・スイッチを使って行います。モノクロディスプレイ用コネクタに出力できるのは次の組合わせです。

(1)　テキスト画面とグラフィック画面のMix画面を出力します。

M　T

(2)　テキスト画面だけを出力します。

M　T

注　意

専用高解像度ディスプレイモードで，カラーディスプレイとモノクロディスプレイを同時に使用する場合は，両方とも〈専用高解像度〉のものを使用する必要があります。

8.5 N-BASICのグラフィック表示

N-BASICは，独立したビデオVRAMを持っていませんから，今まで説明してきたようなグラフィック表示はできません。N-BASICでは，同一画面に，テキスト画面とグラフィック画面を表示しています。ここに，主なN-BASICのグラフィック表示の機能を挙げておきます。

1. 160×100ドットのグラフィック表示です。(キャラクタ単位でカラー指定ができます。)
2. line文があります。(ただし，N88-BASICのline文のような多彩な機能はありません。)
3. 画面操作機能として，get@, put@があります。

詳しくは，「PC-8001ユーザーズマニュアル」か，「N-BASIC REFERENCE MANUAL」を参照してください。

第9章

第9章

グラフィックス

N$_{88}$-BASICは，豊かなビジネスグラフィックスを提供するために便利な機能を数多く備えていますが，それだけに複雑な部分があります。この章では，その中でも重要だと思われる部分に的を絞って説明します。

9.1　グラフィック画面の座標系

9.1.1　オリジナル・スクリーン座標とスクリーン座標，ワールド座標

(1)　オリジナル・スクリーン座標

〈オリジナル・スクリーン座標〉は，ディスプレイ上の固定されている物理的な座標で，ディスプレイのドットと1対1に対応しています。このオリジナル・スクリーン座標は，view文だけで使います。

・グラフィックモード(screen 1)

カラーグラフィックモード(screen 0)

(画面は，640×200ドットとなっています。)

(0, 0)　(639, 0)

(0, 199)　(639, 199)

オリジナル・スクリーン座標(640×200ドット)

・専用高解像度ディスプレイモード(screen 2)

(専用高解像度ディスプレイを使用していない場合は，このように動作しません。)

(0 , 0)　　　　　　　　　　　　　　　　(639, 0)

(0 ,399)　　　　　　　　　　　　　　　　(639,399)

オリジナル・スクリーン座標(640×400ドット)

「オリジナル・スクリーン座標」は固定された，ディスプレイの物理的な座標です。この座標系はview文だけで使用します。

(2)　スクリーン座標

〈スクリーン座標〉は，ビューポート内の座標で，ビューポートの左上スミが必ず原点(0， 0)となります。スクリーン座標系では，座標系の点とディスプレイ画面のドットが1対1に対応しています。このスクリーン座標は，get文，put文，point関数および拡張命令cmd turtle文で使います。詳しくは，次の「9.1.2ウィンドウとビューポート」を参照してください。

例

カラーグラフィックモードでの〈オリジナル・スクリーン座標〉と〈スクリーン座標〉(△はスペースを1個入れることを表します。)

```
screen △ 0,1 ↵
view (100, 50)-(400, 150),, 7 ↵
```

この命令を入力したとき，画面に表示される白い枠が，ビューポートです。このときの各座標は，次のようになっています。

(0, 0)

(100, 50)

ビューポート

(400, 150)

(639, 199)

オリジナル・スクリーン座標

(0, 0)

ビューポート

(300, 100)

スクリーン座標

「スクリーン座標」は，ビューポート内の物理的な座標で，いつも左上スミが，原点(0, 0)となります。put文，get文，point関数および拡張命令cmd turtle文で使います。

(3)　ワールド座標

N_{88}-BASICには，スクリーン座標，オリジナル・スクリーン座標という画面上の座標とは別に，〈ワールド座標〉という架空の座標系を持っています。N_{88}-BASICのほとんどのグラフィック命令は，このワールド座標を用いて座標の指定を行います。ワールド座標の大きさは，(−1.70141E+38, −1.70141E+38)～(1.70141E+38, 1.70141E+38)です。この大きな座標系をいつもディスプレイに表示させるのではなく，その一部を指定して("ウィンドウ"といいます)，ディスプレイに表示させます。詳しくは，次の「9.1.2ウィンドウとビューポート」を参照してください。

(−1.70141E+38, −1.70141E+38)→(1.70141E+38, −1.70141E+38)

(0, 0)

(−1.70141E+38, 1.70141E+38)　(1.70141E+38, 1.70141E+38)

ワールド座標

「ワールド座標」は，N88-BASICの中の架空の座標(論理座標)で，(−1.70141E+38, −1.70141E+38)〜(1.70141E+38, 1.70141E+38)の大きさを持ちます。N88-BASICのほとんどのグラフィック命令が，この座標系で動作します。

9.1.2　ウィンドウとビューポート

ワールド座標は，とても大きな座標系なので，通常は，表示させたいワールド座標のある領域を指定して，その領域をディスプレイ画面のある領域に変換して表示させます。このワールド座標の中で，ディスプレイに表示させる領域を，「ウィンドウ」と呼びます。また，ディスプレイの画面で表示に使われる領域を「ビューポート」と呼びます。この「ウィンドウ」と「ビューポート」は，screen文でグラフィックのモードを設定すると，次の値に初期設定されます。

(1) カラーグラフィックモード(screen 0)，グラフィックモード(screen 1)を指定したとき。

(0, 0)
(0, 0)
ビューポート
ウィンドウ
(639, 199)
(639, 199)

ワールド座標　　　　スクリーン座標

(2) 専用高解像度ディスプレイモード(screen 2)を指定したとき。(ただし，専用高解像度ディスプレイを使用していない場合は，このように動作しません。)

(0, 0)
(0, 0)
ビューポート
ウィンドウ
(639, 399)
(639, 399)

ワールド座標　　　　スクリーン座標

「ウィンドウ」は，window文を用いて指定します。このとき，指定する座標は，ワールド座標を使います。「ビューポート」は，view文で指定します。view文では，オリジナル・スクリーン座標を使います。

例

window文とview文で「ウィンドウ」と「ビューポート」を指定した場合。

```
screen△ 0, 0 ⏎
window(−20000, −20000)−(5000, 5000) ⏎
view(300, 50)−(600, 150),, 7 ⏎
```


ワールド座標　　　　　　オリジナル・スクリーン座標

オリジナル・スクリーン座標　　　　　　スクリーン座標

ですから，グラフィック命令を実行する場合は，次のように考えてください。

(−20000, −20000)

ウィンドウ

(5000, 5000)

(0, 0)

ビューポート

(300, 100)

ワールド座標　　　　　　　　　　スクリーン座標

window文で指定できるウィンドウの大きさには次の制限があります。

(X_1, Y_1)

(X_2, Y_2)

ウィンドウ

$X_1 \sim X_2$, $Y_1 \sim Y_2$ のとる値の範囲は，−1.70141E＋38～1.70141E＋38です。ただし，ウィンドウを設定する場合に， window(X_1, Y_1)−(X_2, Y_2)で用いるX_1, X_2, Y_1, Y_2は,

$$-1.70141E+38 < \frac{X_2 - X_1}{Y_2 - Y_1} < 1.70141E+38$$

を満たしていなければなりません。

また，circle文やline文など，多くのグラフィック命令は，ワールド座標で座標や長さを指定しますが，このとき指定できる座標は，その座標や実行結果が次に示す範囲内に存在するときに限ります。

window(a_1, b_1)−(a_2, b_2)

view(x_1, y_1)−(x_2, y_2)

(a_1, b_1) ウィンドウ (a_2, b_2) — (x_1, y_1) ビューポート (x_2, y_2)

横方向

$$-32768\times\frac{a_2-a_1}{x_2-x_1}+a_1\sim 32767\times\frac{a_2-a_1}{x_2-x_1}+a_1$$

縦方向

$$-32768\times\frac{b_2-b_1}{y_2-y_1}+b_1\sim 32767\times\frac{b_2-b_1}{y_2-y_1}+b_1$$

例

電源を「ON」にしたとき，またはリセットボタンを押したとき。
(screen 0, screen 1の場合も同じ)

(0, 0) ウィンドウ (639, 199) — (0, 0) ビューポート (639, 199)

ワールド座標　　　　オリジナル・スクリーン座標

横方向

$$-32768\times\frac{639-0}{639-0}+0=-32768$$

$$32767\times\frac{639-0}{639-0}+0=32767$$

よって横方向は−32768〜32767で指定します。

縦方向

$$-32768 \times \frac{199-0}{199-0} + 0 = -32768$$

$$32767 \times \frac{199-0}{199-0} + 0 = 32767$$

よって縦方向は，−32768〜32767で指定します。

例

window（−10000，−5000）−（20000，10000）

view（100，50）−（350，150）

（−10000，−5000）

ウィンドウ

（20000，10000）

（100，50）

ビューポート

（350，150）

ワールド座標　　　　　　　　　　スクリーン座標

横方向

$$-32768 \times \frac{20000+10000}{350-100} - 10000 \fallingdotseq -3.94E+06$$

$$32767 \times \frac{20000+10000}{350-100} - 10000 \fallingdotseq 3.92E+06$$

よって，横方向は−3.94E+06〜3.92E+06

縦方向

$$-32768 \times \frac{10000-5000}{150-50} - 5000 = 1.64E+06$$

$$32767 \times \frac{10000-5000}{150-50} - 5000 = 1.63E+06$$

よって縦方向は，−1.64E+06〜1.63E+06となります。

電源を「ON」にした後や，リセットした後，またはscreen文でグラフィックのモードを指定した場合は，ウィンドウは初期設定されますが，view文でビューポートを指定すると，それに合わせて，ウィンドウも変化します。

ですから，ウィンドウを変化させたくない場合はwindow文でウィンドウを指定します。

(a)　window文で指定しない場合

screen△ 0, 0 ⏎

(0, 0)　(0, 0)

ウィンドウ　ビューポート

(639, 199)　(639, 199)

ワールド座標　スクリーン座標

view(150, 50)−(450, 150),,7 ⏎

(0, 0)　(0, 0)

ウィンドウ　ビューポート

(300, 100)　(300, 100)

ワールド座標　スクリーン座標

(0, 0)

(150, 50)

(450, 150)

(639, 139)

オリジナル・スクリーン座標

(b)　window文を指定した場合

screen△ 0, 0 ⏎

(0, 0)　　(0, 0)

ウィンドウ　　ビューポート

(639, 199)　　(639, 199)

ワールド座標　　スクリーン座標

```
window( 0 ,  0 )－(639, 199) ⏎
view(150, 50)－(450, 150),,7 ⏎
```

(0, 0)　　(0, 0)

ウィンドウ　　ビューポート

(639, 199)　　(300, 100)

ワールド座標　　スクリーン座標

(0, 0)

(150, 50)

ビューポート

(450, 150)

(639, 199)

オリジナル・スクリーン座標

例

次の命令を実行してみましょう。

```
screen△ 0, 0 ⏎
circle( 0 ,  0 ), 5000, 1 ⏎
```

画面には何も現われませんね。このとき，ウィンドウとビューポートは先程説明した初期値に設定されていますから，円は，次のように描かれています。

これでは，せっかく描いた円も表示されませんね。そこで，次のwindow文を実行して，ウィンドウを設定し直します。

window(−10000，−10000)−(10000，10000)⏎

そしてもう一度，同じcircle文を実行します。

circle(0，0)，5000，1⏎

すると今度は円が現われましたね。このときのウィンドウとビューポートは，次の図のようにセットされています。

screen文を実行して，ウィンドウを初期値に設定します。

screen△0，0⏎

9．1．3　拡大と縮小（window文とview文の応用）

一画面に数種類のグラフや図形などを表示させたとき，どうしても小さくてわかりにくいものになってしまいます。そんなとき，その一部分を拡大できれば，大変便利ですね。

N88-BASICでは，window文とview文を使うだけで，今描いた小さな図形を簡単に拡大できます。ですから，より大きな図形にするには，最初からプログラムやデータを作り直さねばならないという事はありません。

わかりやすいように，circle文を使って説明します。まず，次の命令を実行しましょう。

```
screen△0, 0 ⏎
circle(479, 50), 30, 6⏎
```

すると画面の右上スミに円が現われましたね。

この円の半径を２倍にしてみましょう。次の命令を実行してみましょう。

```
window(319, 0)-(639, 100)⏎
view(0, 0)-(639, 199)⏎
```

この後，さっきと同じcircle文を実行させます。

```
circle(479, 50), 30, 6⏎
```

すると，倍の半径を持つ円が現われましたね。種明しは，次の図を見てください。

(0, 0)　(0, 0)

ウィンドウ　ビューポート

(639, 199)　(639, 199)

ワールド座標の一部　スクリーン座標

window(319, 0)－(639, 100) ↵

view(0, 0)－(639, 199) ↵

(319, 0)　(0, 0)

(639, 100)

ウィンドウ　ビューポート

(639, 199)

ワールド座標の一部　スクリーン座標

逆に縮小する場合も同じ考えを応用できます。まず，次の命令を実行します。

screen△ 0, 0 ↵

cls△ 2 ↵

circle(320, 100), 200, 3 ↵

(cls△2は，グラフィック画面をクリアする命令です。)これで，画面に大きな円が描けましたね。この円の半径を半分にします。次の命令を実行します。

window(0, 0)－(639, 199) ↵

view(0, 0)－(319, 100) ↵

この後で，もう一度，circle文を実行します。

circle(320, 100), 200, 3 ↵

すると画面の左上に，半径が半分の円があらわれたでしょう。拡大のときと同じ考えですね。下にその図を示しておきます。

window(0, 0)−(639, 199) ⏎
view(0, 0)−(319, 100) ⏎

screen△ 0, 0 ⏎
で元の値に戻しておきます。

9.2 グラフィック画面のカラー指定

グラフィック画面のカラー指定は，カラーグラフィックモードと，グラフィックモードおよび,専用高解像度ディスプレイモードとは,全く異なっています。カラーグラフィックモードでは，640×200ドットの各ドットに対してカラー指定ができます。グラフィックモードと専用高解像度ディスプレイモードでは，テキスト画面のカラー指定を利用して，キャラクタ単位でカラー指定を行います。

9.2.1 バックグラウンドカラー

「8.1.3テキスト画面のカラー指定」の中で，(i)として，「ファンクションコード/フォアグラウンドカラー/バックグラウンドカラーの指定を行うcolor文」の事が説明してありますね。ここでは，テキスト画面に関係がある〈ファンクションコード〉だけ詳しく述べられています。というのも，他のパラメータは，グラフィック画面のカラー指定を行うからです。まず，このcolor文の書式を示します。

color△〔〈ファンクションコード〉〕〔,〈バックグラウンドカラー〉〕
〔,〕〔,〈フォアグラウンドカラー〉〕⏎

〈フォアグラウンドカラー〉は，主に，カラーグラフィックモードにだけ関係あるので，後まわしにします。

〈バックグラウンドカラー〉は,グラフィック画面の背景の色のことです。この値を指定したcolor文を実行した後，cls文で画面をクリアすると，画面の背景がこの色に変わります。また，preset文を，色指定なしで実行すると，この色が採用されます。〈バックグラウンドカラー〉は，パレット番号を用いて指定します。

（参　照）

N88-BASICでは，色のパラメータを指定する場合に,「カラーコード」で指定する場合と「パレット番号」で指定する場合があります。「BASIC REFERENCE MANUAL」の「第1章　カラー」を参照して，カラーコードとパレット番号の関係をよくつかんでください。

9.2.2　カラーグラフィックモードのカラー指定

N88-BASICのグラフィック命令には，パレット番号が指定できるようになっており，これで，カラー指定が行えます。また，このパレット番号を省略した場合は，先に説明したcolor文の〈フォアグラウンドカラー〉で指定した値が，パレット番号として採用されます。もし，指定してなければ，〈フォアグラウンドカラー〉は“7”にセットされていますから，パレット番号を指定しないグラフィック命令は，パレット番号を7として実行します。

カラーグラフィックモードのカラー指定は，パレット番号を使いますから，パレットを変えるとカラーは変ります。例えば，次の命令を実行してみましょう。

screen△ 0, 0 [↵]

circle(100, 100), 50, 1[↵]

これで，青い円が描かれましたね。パレットを変えてみましょう。

color=(1, 2)[↵]

円が赤くなりましたね。もとにもどすには，次の命令を実行します。

color=(1, 1)[↵]

(注　意)

カラーグラフィックモードでは，モノクロディスプレイにグラフィック画面を表示させている場合に，指示した色によっては薄く表示されることがあります。

(参　照)

N88-BASICには，3種類のcolor文があります。それぞれ大切な命令ですから，「BASIC REFERENCE MANUAL」をよく読んでください。

9.2.3 グラフィックモード，専用高解像度ディスプレイモードのカラー指定

グラフィックモードや専用高解像度ディスプレイモードで，グラフィック命令を実行するときは，グラフィック命令のパレット番号では，カラー指定はできません。パレット番号が０の場合は，黒で表示し，０以外のときは，白を表示します。これらのモードでカラー指定を行うときは，テキスト画面のカラー指定を応用します。このとき使用するのが，「8.1.3テキスト画面のカラー指定」の(ii)で説明したcolor文です。もう一度，読み直していただければ，どのように使えば良いか簡単に理解していただけると思います。

例

グラフィックモードで円を描き，後で色を着けます。

```
width△ 80, 25 ↵
screen△ 1, 0, 0, 1 ↵
console△ ,,, 1 ↵
circle(100, 100), 80, 1 ↵
color@(2, 7)-(30, 20), 2 ↵
```

console文でカラーモードを指定しなければ，カラー指定でなく表示モードの指定になり，色は変化しませんから注意してください。(「表示モード」がわからない方は，もう一度「8.1.3テキスト画面のカラー指定」をお読み下さい。)

このcolor文を利用した場合，テキスト画面のカラー指定を利用していますから，指定された領域に表示されているテキスト画面の文字の色も変化します。

9.3 グラフィック画面の操作

画面のある部分の図形をいろいろな位置に移動させたいことがよくあります。これは，9.1で説明したwindow文とview文を組み合わせても可能ですが，これから説明するget@文，およびput@文を使えば，すばやく簡単に行えます。

X_1 X_2 a_1 a_2
Y_1 Y_2 b_1 b_2

この例のように(X_1, Y_1)と(X_2, Y_2)で指定される領域に表示されている図形を(a_1, b_1)と(a_2, b_2)で指定される領域に表示したい場合は，次の命令を実行します。(座標はスクリーン座標です。)

get @ (X_1, Y_1)−(X_2, Y_2), A

put @ (a_1, b_1), A, or

get@文は，指定された領域の画面上のデータを配列Aに読み込みます。

put@文は，逆に，配列Aに記憶されている画面のデータを指定された位置に表示します。(上の例ですと，(X_1, Y_1)が(a_1, b_1)に対応します。(a_2, b_2)は，自動的に決まります。) もちろん，配列Aは，get@文を実行する前にdim文で宣言してなければなりません。

参　照

put@文では，or以外にもxor，andなどのモードを選択できます。各機能及び，dim文については，「BASIC REFERENCE MANUAL」を参照してください。

必要な配列の大きさは，次の式で求めることができます。

?(int((横のドット数+7)¥8)*a*(縦のドット数)+4)¥b+1 ⏎

int：小数点以下を切り捨てて整数にまとめる関数

a：3　カラーグラフィックモード

　　1　グラフィックモード，専用高解像度ディスプレイモード

b：8　倍精度型配列を使用する場合

　　4　実数型配列を使用する場合

　　2　整数型配列を使用する場合

例

カラーグラフィックモードでget@(10，10)−(29，39)，D％を実行する場合(D％は整数型配列を表しています。)

a＝3，横のドット数＝20，縦のドット数＝30，

b＝2(Dが整数型配列だから)

?(int((20+7)¥8)*3*30+4)¥2+1 ⏎

〔=(**3*3*30+4**)/**2+1=274/2+1=**〕**138**

したがって配列の宣言は，

dim D%(137)

となります。(配列の添字は，通常は0から始まります。詳しくは「BASIC REFERENCE MANUAL」の「OPTION BASE」を参照してください。)

9.4 PAINT文

N88-BASICのカラーグラフィックモードでは，キメ細かな，そして色彩やかなグラフィック命令が数多く使用できます。中でも，paint文の機能はすばらしく，基本カラー8色だけでなく，中間色や模様も扱えます。この節では，paint文を自由に使えるように，基本的な事柄から説明していきます。

9.4.1 ビット，バイト，16進数

「ビット」や「バイト」という単位は，2進数(0と1だけで数を表現します。)を使用しているコンピュータを扱う上で，どうしても切り離せないものです。めんどうがらずによく理解してください。

「ビット」とは，2進数における数字の0(なし)または1(ある)のことで，要するに2進法における桁数を表します。

例

10進数の4は，2進数では，100となります。

$4_{(10)}=100_{(2)}$

この$100_{(2)}$は，2進数で3桁，つまり3ビットということになります。

「バイト」は，8ビットを1まとめにして扱うときの単位です。

「16進数」は，逆に，0からF(10進数の15にあたります。)を用いるもので，2進数と10進数の間をとりもつような位置にあります。次の図を参考にしてください。

10進数	2進数	16進数
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	10000	10
17	10001	11
18	10010	12
	⋮	⋮
254	11111110	FE
255	11111111	FF

数が大きくなってくると，２進数では，桁数が大きくなり，また，10進数で表すといくつになるのかと，換算するのも大変です。そこで，２進数ではなく，16進数を使うわけです。２進数を16進数に直すのは，比較的簡単です。

例

101000110100010(2)を16進数に変換する場合。

２進数を下から４ビットごとに区切り，先の対応表と照し合わせて16進数に置き換えます。

```
1 0 1 0 0 0 1 1 0 1 0 0 0 1 0
\___/ \_____/ \_____/ \_____/
  5      1       A       2
```

これで終わりです。

101000110100010(2)＝51Ａ2(16)

9F5B(16)を２進数になおす場合。

先程の逆の操作を行います。

```
   9       F       5       B
/‾‾‾‾‾\ /‾‾‾‾‾\ /‾‾‾‾‾\ /‾‾‾‾‾\
1 0 0 1 1 1 1 1 0 1 0 1 1 0 1 1
```

ですから，

9F5B(16)＝1001111101011011(2)

となります。

解　説

ドット

ドットとは，画面(または，プリンタ等)で表現できる最小の点のことで，N88-BASICの画面表示では，最も細かいモード(専用高解像度ディスプレイモード)で，横640×縦400ドットとなっています。N88-BASICのカラーグラフィックモードでは，横640×縦200ドットです。

9.4.2　中間色が使える理由

ここでは，中間色または模様が扱える原理を簡単に説明しましょう。

N88-BASICにおいても，N-BASICと同様に，扱える色は8色(青，赤，紫，緑，水色，黄色，白，黒)です。これらの色は，光の3原色である青，赤，緑の組み合わせによって作られた色です。(赤と緑を混ぜると黄色，3色を混ぜると白になります。)

緑 4
黄色 6
水色 5
白 7
赤 2
紫 3
青 1

(注)数字はカラーコードまたはパレット番号です。

N88-BASICとN-BASICで扱える色は同じなのに，何故，一方は中間色が扱えるのに，他方は扱えないのでしょうか。

答えは簡単です。N88-BASICは，カラーグラフィックモードを選択すれば640×200ドットの各ドットごとに色指定ができるからです。例えば赤と黄を交互にドット指定すれば，ドットが細かいために人間の目には，あたかも，オレンジ色で塗ったように見えるのです。しかし，N-BASICは，160×100ドットでキャラクタ単位の色指定しかできません。ですから，交互に赤と黄を指定しても中間色にはなりません。ですから，中間色で色を塗るには，ドットごとに色を指定しなければならないことがおわかりいただけたと思います。

9.4.3 paint文

paint文は，指定した色の枠内をある色で塗ることができる命令です。N88-BASICには，２種類のpaint文があります。

(1) パレット番号に従って，８色でペイントする場合。

中間色などは用いず，パレット番号で指定できる色でペイントする場合は，次のpaint文を使用します。

paint(x, y), 〈パレット番号〉, 〈境界のパレット番号〉

ここで，(x, y)は塗り始めるドットのワールド座標で，パレット番号は，０～７(黒～白)までの８種類です。

例

赤い枠の中を水色でペイントする場合。

```
screen△ 0, 0 ⏎
cls△ 2 ⏎
line(50, 50)-(450, 150), 2, b ⏎
line(100, 75)-(200, 100), 2, b ⏎
paint(300, 100), 5, 2 ⏎
```

(△はスペースを１個挿入することを表しています。)

(2) ８色だけでなく，中間色または，模様でペイントする場合。

中間色を用いる場合は，ドットごとに色を指定しなければなりませんから，パレット番号だけでは指定できません。中間色や模様を使ってペイントする場合のpaint文は，次の形になります。

paint(x, y), 〈ドットパターン〉, 〈境界のパレット番号〉〔, 〈バックグラウンドのドットパターン〉〕

〈ドットパターン〉および,〈バックグラウンドのドットパターン〉は,文字列で与えます。

〈ドットパターン〉は，通常，次のように指定します。

chr$(&h XX)+chr$(&h XX)+chr$(&h XX)

(青の要素)　(赤の要素)　(緑の要素)

この"XX"を，先程の２桁の16進数で指定します。(chr$と&hについては「BASIC REFERENCE MANUAL」と参照してください。ここでは，おまじないと思って先に進んでもかまいません。)では，まず，実際に

中間色を使ってみましょう。

例

白で描かれた円の中をオレンジ色で塗りつぶす場合。

```
screen△ 0, 0⏎
cls△ 2⏎
tile$=chr$(&h 00)+chr$(&h ff)+chr$(&h 55)⏎
circle(320, 100), 100, 7⏎
paint(320, 100), tile$, 7⏎
```

この例では，ドットパターンをtile$という文字変数に代入してから，paint文を実行しています。実際にディスプレイの画面の各ドットで考えてみましょう。16進数と2進数を思い出してください。

	2進数	16進数
青の要素	0 0 0 0 0 0 0 0	0 0
赤の要素	1 1 1 1 1 1 1 1	F F
緑の要素	0 1 0 1 0 1 0 1	5 5

0はその要素がないことを表し，1はその要素があることを表しています。ですから，ディスプレイ上の8ドットを取りあげると，

青の要素	0	0	0	0	0	0	0	0	(00)
	＋	＋	＋	＋	＋	＋	＋	＋	
赤の要素	1	1	1	1	1	1	1	1	(FF)
	＋	＋	＋	＋	＋	＋	＋	＋	
緑の要素	0	1	0	1	0	1	0	1	(55)
	‖	‖	‖	‖	‖	‖	‖	‖	
	赤	黄	赤	黄	赤	黄	赤	黄	

となります。(「9.4.2中間色が使える理由」のカラーコードおよびパレット番号の関係を表す図を参照してください。)この8ドットをずっと並べて表示しますから，赤と黄が混ざって見えて，あたかもオレンジ色のように

見えるわけです。ここまでわかったところで，先程の例の“00”，“ff”，“55”を適当な16進数に変えて入力してみてください。いろいろな色や模様が表示できるでしょう。

ここまでは，8ドットのドットパターンを1行だけ指定して，それを繰り返して色を塗っていたわけですが，複数の行のドットパターンを指定するともっと複雑な模様も表示できます。先程の例を使いますと，

```
tile $=chr $(&h XX)+chr $(&h XX)+chr $(&h XX)
tile $=tile $+chr $(&h XX)+chr $(&h XX)+chr $(&h XX)
tile $=tile $+chr $(&h XX)+chr $(&h XX)+chr $(&h XX)
                ⋮
tile $=tile $+chr $(&h XX)+chr $(&h XX)+chr $(&h XX)
```

というように，ドットパターンを指定します。

例

白い円の中を模様で塗りつぶす場合。

```
screen△ 0, 0 ⏎
cls△ 2 ⏎
tile $=chr $(&h 12)+chr $(&h 34)+chr $(&h 56) ⏎
tile $=tile $+chr $(&h 78)+chr $(&h 9a)+chr $(&h bc) ⏎
tile $=tile $+chr $(&h de)+chr $(&h f0)+chr $(&h 12) ⏎
circle(320, 100), 100, 7 ⏎
paint(320, 100), tile $, 7 ⏎
```

〈バックグラウンドのドットパターン〉は省略して使ってきました。〈バックグラウンドのドットパターン〉の使い方については，「BASIC REFERENCE MANUAL」を参照してください。

グラフィックモードおよび専用高解像度ディスプレイモードの場合のpaint文。

この場合は，青，赤，緑の各要素の指定が1つになり，

```
chr $(&h XX)
```

で1行分のドットパターンを指定します。このとき，1ならば白，0ならば黒が表示されます。

（例）

&haa	1	0	1	0	1	0	1	0
	‖	‖	‖	‖	‖	‖	‖	‖
	白	黒	白	黒	白	黒	白	黒

```
screen△ 1, 0, 0, 1⏎
cls△ 2⏎
tile $=chr $(&h aa)⏎
circle(200, 100), 80, 1⏎
paint(200, 100), tile $, 1⏎
```

9.5　漢字の表示

ディスプレイに漢字を表示させてみましょう。

漢字は，グラフィック画面に表示されます。漢字を表示させる命令は，put文で，次の形式で入力します。

put(X, Y), kanji(〈漢字JISコード〉), {PSET | PRESET},
〈フォアグラウンドカラー〉, 〈バックグラウンドカラー〉

〈フォアグラウンドカラー〉と〈バックグラウンドカラー〉については，「9.2 グラフィック画面のカラー指定」で説明してあります。また，PSET, PRESETについては，「BASIC REFERENCE MANUAL」を参照してください。

〈漢字JISコード〉は，「付録7 漢字コード表」に書いてある数字です。この表では，16進数で表してありますから，数字の前に&hをつけます。

Note **漢字を出力するput文は，N88-BASIC(N88DISK-BASIC)だけの命令です。**

（例）

「亜」という字を表示させたい場合。

〈漢字JISコード〉は，&h 3021となります。

screen△ 0, 0 [↵]
cls△ 2 [↵]
put△ (100, 100), kanji(&h 3021), pset, 7, 0 [↵]

では，次に漢字ROMに入っている全文字を表示させてみましょう。まず，次のプログラムを入力します。

screen△ 0, 0 [↵]
cls△ 3 [↵]
new [↵]
[CAPS]

```
100  ' X : ヒョウジイチ (X )
110  ' Y : ヒョウジイチ (Y )
120  ' A : カンジコード (A )
130  SCREEN 0, 0: CONSOLE , ,0
140  CLS 3
150  A=&H20 : B=&HFF
160  FOR Y=0 TO 180 STEP 18
170  FOR X=2 TO 622 STEP 20
180  PUT (X, Y) , KANJI (A) , PSET, 7, 0
190  IF A=B THEN 200 ELSE A=A+1: GOTO 260
200  IF A=&HFF THEN CLS 2: A=&H100: B=&H1FF: GOTO 160
210  IF A=&H1FF THEN CLS 2: A=&H2120: B=&H217F: GOTO 160
220  IF A=&H277F THEN CLS 2: A=&H3020: B=&H307F: GOTO 160
230  IF A=&H4F53 THEN 140
240  A=A+&HA1: B=A+&H5F
250  IF B=&H4F7F THEN B= &H4F53
260  NEXT X,Y
270  GOTO 160
```

では，実行させます。このプログラムは，[STOP] キーを入力すれば止ま

ります。

　　f・5 （またはrun ⏎）

　ストップさせるときは

　　STOP

を入力します。

　このプログラムは，「付録7　漢字コード表」の順に，漢字を画面に表示し続けます。

注　意

　　漢字のフォントは16×16ドットで表されていますので文字の一部が省略されているものがあります。

第10章

第10章
入出力装置

10. 1　ファイル

N_{88}-BASICは，入出力装置との情報のやりとりを「ファイル(書類)」という概念で行います。ですから，目的の「ファイル」は，どこの棚にある何という名前のものかを指定しなければなりません。この指定を行うのが，“ファイルディスクリプタ”と呼ばれるものです。ファイルディスクリプタは，棚にあたる〈入出力装置〉を指定する“デバイス名”と〈書類の名前〉にあたる“ファイル名”で構成されています。

〔ファイルディスクリプタ〕＝〔デバイス名〕＋〔ファイル名〕

10. 1 . 1　ファイル名

ファイル名は，最大で「6文字＋拡張子3文字」で構成されています。拡張子はファイルの内容を識別するために用います。例えば，N_{88}-BASICで書かれたプログラムには，6文字以下のプログラム名の後に，拡張子として，“.n88”を続けると決めておきます。“test 1”というプログラムでしたら，

test 1 . n88

となります。N-BASICで書かれたプログラムには“.n80”をつければ，

test 2 . n80

となります。こうしておくと，フロッピィディスクに記録されているファイル名を表示させた場合など，どのファイルが，どんな形式で記録されているのか一目でわかりますね。しかし，同じ形式のものしか記録しない場合は，9文字まで，プログラム名として使用してもかまいません。

注　意

カセットテープを使用する場合は，「拡張子３文字」は使用できません。すなわち，ファイル名は最大６文字になります。また，RS-232Cポートを入出力装置として選んだ場合は，ファイル名は〈書類の名前〉ではなく，RS-232Cポートの仕様を定義するために使用します。詳しくは，「第14章ターミナルモード」を参照してください。

10.1.2　デバイス名

デバイス名は，次の表のように，各入出力装置につけられています。

入出力装置名	デバイス名	入力	出力
キーボード	**KYBD:**	○	×
スクリーン	**SCRN:**	×	○
プリンタ	**LPT1:** または **LPT:**	×	○
カセットテープ（1200ボー）	**CAS1:** または **CAS:**	○	○
（600ボー）	**CAS2:**	○	○
フロッピィディスク　1	**1:**	○	○
2	**2:**	○	○
3	**3:**	○	○
4	**4:**	○	○
5	**5:**	○	○
6	**6:**	○	○
7	**7:**	○	○
8	**8:**	○	○
RS−232Cポート	**COM1:** または **COM:**	○	○

注　意

“KYBD:”と“SCRN:”は N88DISK-BASICを使用している場合だけ，指定できます。

デバイス名に含まれる“1”および，フロッピィディスクの“1：”は省略できます。フロッピィディスクのデバイス名については，「12.1接続ディスクとデバイス名」を参照してください。

また，フロッピィディスクユニットを接続していない場合は，デバイス名を省略すると「cas：(cas1：)」が指定されたと見なされます。この時，「cas2：」は省略できません。(「2：」と指定してもエラーとなります。)

10.2　ファイルのOPEN, CLOSE

ファイルディスクリプタで，「どこの棚のどのファイル」なのかを指定できましたね。そのファイルを実際に扱うにはどうすればよいでしょうか。新しいファイルを作成したり，追加・削除するためには，ファイルを開いたり，使い終わったファイルを閉じたりする動作が必要になってきます。この動作を行う命令が，open文とclose文です。open文では，ファイルディスクリプタを指定するだけでなく，そのファイルに対して，どのような処理を行うかを宣言します。この宣言を〈モード〉といいます。

モード	処　　理
INPUT	既につくられているファイルから，データを読み込みます。
OUTPUT	新しいファイルを作成して書き込みます。
APPEND	既にあるファイルに，続けてデータを書き込みます。

(解　説)

〈モード〉

N88-BASICでは，〈モード〉は単にファイルのポインタの最初の位置を決定します。

INPUT　ポインタの最初の位置はファイルの始めで，ファイルが見つからない場合は，エラーになります。

OUTPUT　最初の位置はファイルの始めで，常に新しいファイルを作ります。

APPEND 最初の位置は，ファイルの終りで，ファイルが見つからない場合は，エラーになります。なお，カセットテープ上のファイルをAPPENDモードでオープンすることはできません。

for 〈モード〉が省略された場合には，ポインタの最初の位置はファイルの先頭になります。もし，ファイルが見つからない場合は，その名前のファイルができます。

ファイルの開閉は，open, close文で行います。

ここで少しむずかしくなりますが，ファイル番号について少し説明しておきます。N88-BASICは，ファイルとの入出力を行うとき，バッファという窓口を設けます。つまり，ファイルにデータを書き込むとき，あるいは逆に読み出すときは，この窓口を通じて行います。この窓口は，同時に1つのファイルにしか使用できません。ですから1度，オープンさせた窓口は，1度閉じなければ，他のファイル用にオープンすることはできません。この窓口の番号が，ファイル番号なのです。ですから，open文で，ファイルディスクリプタにファイル番号を指定すると，以後は，ファイル番号でファイルを指定します。ファイル番号は，＃1から，「How many files?」で指定した値まで指定できます。

例

カセットテープにファイルをオープンさせ，データを書き込む場合。

```
10 open "cas : tape" for output as ＃1
        ⋮
100 print ＃1, data
```

"＃1"がファイル番号で，print ＃1では，＃1の窓口，すなわちカセットテープのファイルに"data"というデータを出力します。(カセットテープへの，データのセーブについては，「11.2データのセーブ，ロード」を参照してください。)

注　意

フロッピィディスクにファイルをOPENしている状態で，そのフロッピィディスクを取り出すと，そのフロッピィディスクの内容を壊して

しまう場合があります。フロッピィディスクを取り出す場合は，必ずclose文またはend文を実行してください。

10. 3　How many files?

N88-BASIC(N88DISK-BASIC)では，スタートするとすぐに

How many files?

と尋ねてきます。これは，「同時にいくつのファイルを使用しますか」，または別の表現をすると，「ファイルとのデータの入出力用に，いくつの窓口(バッファ)を準備すれば良いのですか」と問い同わせてきているのです。N88-BASICでは，最大15まで指定できます。しかし，ここで「同時に」という点に注意してください。例えば，file 1, file 2, file 3, file 4という4つのファイルを扱うプログラムを使うとします。このとき，file 1とfile 3は，常に使えなければならないが，file 2とfile 4は，交互に使用する場合は，この2つのファイルで1つの窓口を使います。つまりfile 2を使用する場合は，file 2として窓口をオープンし，使い終わったら，すぐにクローズして，同じ窓口をfile 4としてオープンします。こうすれば，4つのファイルを扱うのに3つの窓口で済むわけです。

例

```
How many files? 3
          ⋮
100 open "1 : file 1" for input as #1
200 open "cas : file 2" for output as #2
300 open "2 : file 3" for input as #3
          ⋮
1000 close #2
1100 open "2 : file 4" for input as #2
          ⋮
```

「ファイルを4つ使うならば，窓口も4つ準備すればよい」と思われる方があるかもしれませんね。しかし，窓口は，ユーザーズメモリの中に作

られますから，不必要に大きく指定すると，それだけ，使えるメモリ領域が少なくなってしまいます。

How many files?と尋ねてきたとき，数値を指定せず，リターンキー(⏎)を入力すると，フロッピィディスクユニットが接続されていない場合は1が，接続されている場合はそのドライブ数が設定されます。また数値以外のものを入力すると0が入力されたとみなされます。

10.4　キーボードとスクリーン

N88DISK-BASICでは，キーボードとディスプレイをファイルとして扱う事ができます。この機能を利用すると，フロッピィディスクやプリンタを扱うプログラムを作成する場合に非常に便利です。

例えば，フロッピィディスク上のファイルからデータを読み込んで処理を行い，再びフロッピィディスクの別のファイルに書き込むプログラムを作成するとします。この場合に，最初からフロッピィディスクを用いてプログラムを作成するよりも，キーボードからデータを入力し，スクリーンへデータを表示させればより簡単に作成できますね。そこで，プログラムを作成している間は次のようにファイルをオープンします。

```
100 open "kybd : test 1" for input as #1
200 open "scrn : test 2" for output as #2
        :
1000 input #1, a$
        :
2000 print #2, b
        :
```

プログラムが完成すれば，行番号100と200の2行を変更すればフロッピィディスク用のプログラムになります。(行番号1000や2000の入出力文は変更しなくても良いわけです。)

```
100 open "1 : test 1" for input as #1
200 open "2 : test 2" for output as #2
        :
```

注　意

キーボード(KYBD：)およびスクリーン(SCRN：)では，使用できない入出力命令がありますから注意してください。(各入出力命令については「BASIC REFERENCE MANUAL」を参照してください。)

第11章

第11章
カセットテープの使い方

PC-8801MKIIでは，本体の電源を切ったり，リセットボタンを押すと，あなたが入力したプログラムやデータは，すべて消えてしまいます。ですから，再び使用する可能性のあるプログラムやデータは，フロッピィディスクやカセットテープに保存しておきます。そうすれば，電源を切ったり，リセットボタンを押したりするたびに，最初からプログラムやデータを入力する必要がなくなり，フロッピィディスクやカセットテープから読み込めばよいわけです。このように，メモリの中のプログラムやデータをフロッピィディスクやカセットテープに書き込むことを，「セーブ」するといい，逆にプログラムやデータをフロッピィディスクやカセットテープからメモリへ読み込むことを「ロード」するといいます。

カセットからプログラムをロードするときの注意事項

カセットからプログラムをロードする場合，もしディップスイッチ SW1の6 がオフのときはオンにしてリセットボタンを押してからロードを行うか，又は“NEW ON &H20”を実行したあとロードを行ってください。

11.1 プログラムのロード，ベリファイとセーブ

カセットテープへのロードとセーブは，どのBASICモードでも実行できますが，ここでは，N88-BASICモードの場合を説明していきます。使用するカセットテープレコーダーは，一般用オーディオカセットテープレコーダーです。(以後，テープレコーダと略します。)テープレコーダとしては，パーソナルコンピュータ用に設計されたデータレコーダ(PC-DR311など)が操作しやすく，失敗も少ないようですが，家庭用のものでも充分機能が果せます。では順を追って説明します。

11.1.1 PC-8801MKIIとテープレコーダとの接続

「第4章本体と周辺装置との接続」に従ってPC-8801MKⅡとカセットテープレコーダを接続します。このとき，次の点に注意してください。

(1) リモート端子(黒色)を接続した場合

リモート端子を接続しますと，テープレコーダのON/OFFは，PC-8801MKⅡがコントロールします。ですから，PLAYボタンを押していても，PC-8801MKⅡで，カセットテープを使用する命令が実行されなければ，カセットテープは回りません。この機能によって，必要な間だけ，カセットテープを回すことができ，カセットテープの無駄使いが防げ，また，手間もかかりません。

(2) リモート端子を接続しなかった場合

この場合は，テープレコーダのスタート，ストップは，すべて自分でやる必要があります。ですから，カセットテープを使用する命令を実行する前にPLAYボタンを押し，終わるとSTOPボタンを押すという操作が必要になります。

普通，1つのプログラムをロードしたり，セーブする場合は，リモート端子を接続せずに操作することが多いようですが，プログラムの中でデータのロード，セーブを行う場合は，リモート端子を接続していないと，うまく動作しない場合があります。以後，リモート端子が接続されているとして説明をします。リモート端子を接続しない場合は，先に「11.4リモート機能を使用しない場合」を読んでください。

11. 1 . 2　プログラムのセーブ

本体とテープレコーダが接続できたら，キーボードから，次のプログラムを入力してください。(入力が終わっても実行させないでください。)

```
new ⏎
100 input a, b, c, d ⏎
200 print "Set tape, then push PLAY and REC" ⏎
300 input "ok"; a$ ⏎
400 open "cas : suchi" for output as ＃1 ⏎
500 print ＃1, a, b ⏎
600 print ＃1, c, d ⏎
700 close ＃1 ⏎
800 end ⏎
```

入力したプログラムがまちがっていないかどうか，ディスプレイとよく見比べてください。まちがっていなければ，このプログラムをカセットテープへセーブしましょう。

1. 録音してもよいカセットテープをテープレコーダにセットします。プログラムやデータを入力すると，当然，前に録音してあったものは消えてしまいます。また，このとき，テープカウンタを0にセットしておけば，どこからセーブしたのかすぐにわかりますね。
2. テープがセットできたら，録音ボタン(REC)とPLAYボタンを押します。リモート端子(黒)が接続してあれば，カセットテープは回り始めないはずです。そこで次の命令を入力します。

   motor ⏎

   ⏎ を入力すると同時に，“カチッ”という音がして同時にテープが回り始めます。数秒たったらもう一度，

   motor ⏎

   を入力します。すると再び“カチッ”という音がして，テープは止ります。これでセーブの準備が整ったわけです。

注　意

カセットテープには，始めと終わりにクリアフィーダという磁性体が塗ってない部分があります。この部分には記録できません

から，プログラムやデータをセーブする前にスキップしておく必要があります。また，前のプログラムやデータとの区別をつけるためにも，セーブする前は，motor文でテープを送る習慣をつけましょう。

motor文は，モーターをコントロールします。
motor ⏎ ⇨モータースタート⇨motor ⏎ ⇨モーターストップ

3.　いよいよプログラムをセーブします。PLAYボタンと録音ボタンは押したままですね。そこで，次の命令をキーボードから入力します。

save "cas：test" ⏎

4.　モーターが回り始め，しばらくすると"OK"が表示され，モーターがストップしてプログラムのセーブは終わります。

ここで使用したsave文が，プログラムを「セーブ」する命令です。

save "〈ファイルディスクリプタ〉" ⏎

今の場合ですと，ファイルディスクリプタは"cas：test"ですね。「cas：」がデバイス名で「test」がファイル名です。

解　説

デバイス名「cas：」

N88-BASICのカセットテープのデバイス名には「cas：(又はcas1：)」と「cas2：」の２つがあります。この２つのデバイス名は，PC-8801MKⅡとテープレコーダとのデータの転送速度を指定するために準備されています。詳しくは，「11.3カセットテープの転送速度」をお読みください。

カセットテープへのプログラムのセーブはsave文で行います。

save "cas：XXXX"(1200ボー)

又は，

save "cas2：XXXX"(600ボー)

11.1.3 ベリファイ

カセットテープにプログラムをセーブした場合は，必ず正しくセーブされたかどうかチェックします。つまり，メモリの中のプログラムとカセットテープにセーブされたプログラムが等しいかどうか比べるわけです。この操作を“ベリファイ”といいます。ではベリファイの実行のしかたを順を追って説明しましょう。

1. プログラムのセーブを始めたところまでテープを巻き戻します。リモート端子を接続している場合はmotor文を使います。巻き戻しが終わったら，モータストップ(motor ⏎)を実行しておきます。
2. テープレコーダの音量と音質を調整します。音量は，普段使用する場合よりもやや大きめにセットし，音質が調整できる場合は，高音が強調される側へセットします。
3. PLAYボタンを押した後で，次の命令を実行します。

   load? “cas：test”

4. ”test”というプログラムが見つかると

   Found：test

   と表示し，読み込んだデータとメモリの内容をチェックします。
5. カセットテープにセーブされている内容が一致した場合は，

   OK

   と表示し，一致しない場合は，

   Bad

   と表示してきます。

   一致しなかった原因としては，次の場合が考えられます。

   (1) 再生用の音量と音質の調整が，適当でなかった。

   この場合は，音量と音質を調整し直して，何度か挑戦してみてください。使用しているテープレコーダによっては，音量が，ある特定の範囲でしか，うまく動作しない可能性があります。

   (2) セーブがうまくできていない場合。

   音量，音質をいくら調整してもBadと表示される場合は，セーブした時に，失敗した可能性があります。その原因としては，次のような事が考えられます。

① ケーブルの接続が正しくなかったり，接続プラグが，しっかりと差し込まれていなかった。

② 録音音量が調整できるテープレコーダを使用したが，その調整が正しくなかった。

③ セーブするときに，カセットテープのクリアフィーダの部分をスキップするのを忘れていた。

それぞれ，もう一度よく確認してから，セーブを行ってください。これらの原因を検討してもセーブできない理由がわからない場合は，お近くのBit-INNもしくは，お買い求めの販売店で御相談ください。

Note **プログラムをカセットテープにセーブした場合は，必ずベリファイを実行して，確認しましょう。**

11.1.4 ロード

今度は，カセットテープにセーブしたプログラムを，再び本体のメモリへロードしてみましょう。今はまだ，メモリにプログラムが残っていますから，

[**f・4**] [↵]（または，list[↵]）

を入力すれば，メモリに入っているプログラムが表示されますね。そこでまず最初に，メモリに入っているプログラムを消します。

new[↵]

この命令を実行すると，あなたが入力したプログラムは，すべて消えてしまいます。ですから，

[**f・4**] [↵]（またはlist[↵]）

を入力しても，プログラムは表示されません。では，ロードを始めます。

1. カセットテープを巻き戻します。（リモート端子を接続している場合は，motor文を入力しないとモータは回りませんね。）
2. 音量と音質を確認します。（ベリファイしたときと同じですね。）
3. PLAYボタンを押します。（テープは回りませんね。）
4. 次の命令を入力します。

   load "cas : test"[↵]

5. [↵]を入力するとカセットテープが回り始めます。プログラムが見

つかると(今の場合は「test」ですね。)

Found ：test

とディスプレイに表示し,ロードしている間は,画面の右上スミに“＊”を表示します。指定した以外のプログラム名が見つかると,

Skip：XXXX

と，プログラム名を表示して，次のプログラムを読みに行きます。

6. ロードが無事に終了すれば,

OK

が表示され，カセットテープは止まります。

もし，ロードが途中で失敗すれば，ディスプレイに,

?TP Error

というメッセージが表示され，ブザー(Beep音)が鳴ります。エラーとなった場合には，ベリファイのところで説明した原因を１つ１つチェックしてみましょう。何度，ロードしてもエラーとなる場合は,テープにキズがある場合などが考えられます。また,ベリファイを行っていなければ，うまくセーブができていない可能性があります。このような場合は，もう一度，セーブからやり直す必要があります。

ロードがうまく行けば,

[f・4] [⏎] (またはlist [⏎])

でプログラムが表示されます。

(注　意)

ロードがうまく行かなかった場合に,

[f・4] [⏎] (またはlist [⏎])

を実行すると,途中までのプログラムが表示される場合がありますが,このプログラムを実行させたり，新たな行をつけ加えて実行させることはできません。new文を実行して，最初から入力してください。

また，ロード後“Syntax Error”(又は“SN Error”)と表示され，list コマンドで内容を見ることはできるが，実行すると同じように“Syntax Error”(又は“SN Error”)が表示される場合は,プログラム中に使用されているラベルにBASICの予約語(PRINTとかSINなど)が使われています。まず，ラベルを調べて予約語をラベル名として使っている行を訂正してから実行してみてください。(ラベルについては，「BASIC RE-

FERENCE MANUAL」の 「1.23ラベル(N88-BASIC)」を参照してください。

11.2 データのセーブ，ロード

プログラムのセーブ，ロードがわかったので，次は，データのセーブ，ロードの方法を説明します。この説明では，先程，プログラムのセーブ，ロードに使ったプログラムを使います。

11.2.1 データのセーブ

1. プログラムのセーブとロードに用いたプログラムをロードします。
2. このプログラムを実行させます。

   [f・5] (または，run [↵])
3. すると画面に，"?"を表示します。これは，行番号100のinput文が，キーボードからの入力を要求しているのです。(input文については，「BASIC REFERENCE MANUAL」を参照してください。)

   1, 2, 3, 4 [↵]

   と入力します。
4. すると今度は，次のメッセージが表示されます。

   Set tape, then push PLAY and REC

   OK?
5. そこで，データを入力してもよいカセットテープをテープレコーダにセットします。このとき，テープカウンタを0にしておけば，後でどこからデータが入っているか一目でわかり，とても便利です。
6. カセットテープのセットが終わったら，テープレコーダの録音ボタンとPLAYボタンを押します。これらの準備が終わったら，

   [↵]

   を入力します。
7. [↵] を入力すると，すぐに，カセットテープが回りはじめ，しばらくして，ディスプレイに"OK"と出力し，カセットテープは止まります。

これで，あなたが今，キーボードから入力した数値が，カセットテープにセーブされたわけです。

では，このプログラムを簡単に説明しましょう。

```
100 input a, b, c, d
200 print "Set tape, then push PLAY and REC"
300 input "ok" ; a$
400 open "cas : suchi" for output as ＃1
500 print ＃1, a, b
600 print ＃1, c, d
700 close ＃1
800 end
```

(1) キーボードから数値を入力し，カセットテープのセットも終り，録音ボタンとPLAYボタンを押して，リターンキー(⏎)を入力すると，プログラムは次に，ファイルのオープンを行います。(行番号400)

(2) open文が実行されると，カセットテープへのデータを出力するための窓口(バッファ)が割り当てられ，データのセーブのための準備が整います。

(3) print文を用いてデータをセーブします。(行番号500，600)

print文は，ディスプレイに文字を表示させるだけではなく，このように〈ファイル番号〉を指定すると，ファイルへのデータの出力を行います。

(4) データのセーブが終わったら，close文でファイルを閉じておきます。(行番号700)

11.2.2 データのロード

11.2.1でセーブしたデータをロードしてみましょう。

1. 次のプログラムを入力します。

```
new ⏎
100 a=0: b=0: c=0: d=0 ⏎
200 print "push PLAY" ⏎
300 open "cas: suchi" for input as #1 ⏎
400 input #1, a, b ⏎
500 input #1, c, d ⏎
600 print a, b, c, d ⏎
700 close #1 ⏎
800 end ⏎
```

このプログラムの意味は，おおよそ理解できますね。データのセーブのプログラムとの主な違いは，ファイルをオープンするときに，データの入力用として，"input"を宣言している事と，データのロードを"input #1"で行っている点ですね。

2. プログラムが入力できたら，データをセーブし始めたところまで，カセットテープを巻き戻します。

3. カセットテープの準備ができたら，実行させます。

   [f・5]（または，run ⏎）

4. "push PLAY"というメッセージが現われたら，PLAYボタンを押します。カセットテープが回りはじめて，しばらくすると，ロードされたデータがディスプレイに表示されます。

11.2.3　データの型と数の一致

カセットテープにデータをセーブ，ロードする場合の注意事項として，セーブするときとロードするときの〈データの型と数の一致〉があります。データとして文字をセーブしたのに，それを数値データとしてロードすることはできません。また，1つのprint文で4つのデータをセーブしているのに，input文では2つしかロードしないというわけにもいきません。ただし，データの型と数が一致していれば，ロードしてきたデータを代入する変数は，セーブしたときに使った変数と異なっていてもかまいません。

〔良い例1〕

セーブするとき

```
10 open "cas : sample" for output as #1
            :
100 print #1, a, b, c
            :
500 close #1
```

ロードするとき

```
10 open "cas : sample" for input as #1
            :
150 input #1, X, Y, Z
            :
300 close #1
```

データの型(数値)，数(3つずつ)が一致しているのでOKですね。

〔良い例2〕

セーブするとき

```
10 open "cas : sample" for output as #1
            :
100 print #1, yama $, kawa $
            :
500 close #1
```

ロードするとき

```
10 open "cas : sample" for input as #1
            :
200 input #1, sora $, umi $
            :
600 close #1
```

これも，データの型(文字)，数(2つずつ)が一致しているのでOKです。

〔悪い例〕

セーブするとき

```
10 open "cas : sample" for  output as #1
            :
100 print #1, a, b, c, d,
            :
700 close #1
```

ロードするとき

```
10 open "cas : sample" for input as #1
            :
200 input #1, a, b
210 input #1, c, d
            :
800 close #1
```

データの型(数値)は一致していますが，1つのprint文で4つのデータをセーブしたのに対し，ロードするときは，1度に2個しか扱っていません。これでは，ロードできません。

データのセーブとロードには，データの型と数の一致が必要です。

注　意

カセットテープ上にデータファイルをオープンして入出力を行う場合は，フロッピィディスクの場合と同じようにファイル名を付けることが可能ですが，フロッピィディスクのようにファイル名によって

ファイルを識別することはできません。

従って,

```
OPEN "CAS:A" FOR OUTPUT AS #1
```

として,オープンし,データを書き込んだファイルを,

```
OPEN "CAS:B" FOR INPUT AS #1
```

として,オープンして,データを読み出すこともできます。

11.3 カセットテープの転送速度

N88-BASICでカセットテープを使用する場合に,「600ボー」と「1200ボー」の2つの転送速度を使用します。この指定は,デバイス名を指定すると,N88-BASICが自動的に設定してくれます。

デバイス名	転送速度
cas: (cas1:)	1200ボー
cas 2:	600ボー

同じ長さのプログラムやデータをセーブしたりロードする場合,1200ボーですと,600ボーの半分の時間で実行できます。しかし,600ボーでセーブしたプログラムやデータを1200ボーでロードすることはできません。また,逆に1200ボーでセーブしたプログラムやデータを600ボーでロードすることもできません。ですから,プログラムやデータをセーブする場合は,どちらの転送速度でセーブしたのかを記録しておいて下さい。

注　意

同じテープレコーダで,セーブ,ロードを行う場合は,「1200ボー」で行う方が速く実行できますが,セーブしたテープレコーダと,ロードするテープレコーダが異なる場合に,「1200ボー」ですと,モーターの回転速度の誤差などから,エラーが発生する可能性があります。このような場合は,「600ボー」でセーブ,ロードを行ってください。

(注　意)

PC-8001のカセットインタフェースは,「600ボー」となっています。ですから，PC-8001でセーブされたプログラムやデータをPC-8801MKⅡのN88-BASICモードでロードする場合は，デバイス名は「cas 2：」として「600ボー」でお使いください。

ただし，N88-BASICとN-BASICとは異なる部分があるため，N-BASICのプログラムをN88-BASICでロードできても，listコマンドで内容を見ると，N-BASICでそのプログラムをロードした場合と異なることがあります。N-BASICでセーブしたプログラムをN88-BASICでロードし，N88-BASIC用に訂正したい場合は，一度，N-BASICモードでプログラムをロードし，各行をコメント文(REM文)にしてから，もう一度カセットにセーブし，N88-BASICモードに切り替えてからロードするなど，特殊な手続きが必要です。

11.4　リモート機能を使用しない場合

PC-8801MKⅡとテープレコーダを接続する場合，リモート端子を接続しておくと，テープレコーダのモータの「ON」「OFF」は，PC-8801MKⅡがコントロールします。しかし，リモート端子が接続できない場合は，自分でコントロールしなければなりません。

1.　セーブするとき

PLAYボタンと録音ボタンを押した後，セーブを行う命令を実行します。

2.　ロードするとき

ロードを行う命令を実行したあと，PLAYボタンを押します。

ただし，データのロード，セーブでは，リモート機能を使用しないと実行できない場合がありますから注意してください。

Note **カセットテープを用いて，データのセーブ，ロードを行う場合は，リモート端子を接続して使用しましょう。**

第12章

第12章
フロッピィディスク

この章では，N88DISK-BASICモードを中心にフロッピィディスクの使い方について説明します。

12.1　接続ディスクとデバイス名

「10.1ファイル」で，ファイルディスクリプタは，デバイス名とファイル名で構成されていることを述べました。

〔ファイルディスクリプタ〕＝〔デバイス名〕＋〔ファイル名〕

N88DISK-BASICでは，次の表のように８ドライブまで管理することができます。

入出力装置名	デバイス名
フロッピィディスク　1	1:
2	2:
3	3:
4	4:
5	5:
6	6:
7	7:
8	8:

デバイス名は，接続するディスクの組み合わせによって，次のように対応します。

ミニフロッピィディスクユニット				標準フロッピィディスクユニット			
PC-8801MKII		PC-80S32		PC-8881 PC-8881(1)		PC-8882 PC-8882(1)	
ドライブ1	ドライブ2	ドライブ3	ドライブ4	ドライブ1	ドライブ2	ドライブ3	ドライブ4
1:							
1:	2:						
1:	2:	3:	4:				
3:				1:	2:		
3:	4:			1:	2:		
3:	4:	5:	6:	1:	2:		
5:				1:	2:	3:	4:
5:	6:			1:	2:	3:	4:
5:	6:	7:	8:	1:	2:	3:	4:

注　意

フロッピィディスクユニットのドライブ番号と，デバイス名は，接続されるフロッピィディスクユニットによっては，必ずしも一致していません。

デバイス名の“1：”は省略することもできます。

例

save “1：prog”→save “prog”

N88DISK-BASICで，指定するドライブ番号は，デバイス名での番号です。

デバイス名の指定には，十分気をつけて下さい。特に，プログラムやデータをセーブするときや，ファイルの削除(kill文)のときに，デバイス名を間違えて，ファイルを壊してしまうということのないように注意して下さい。

注　意

作成したプログラムを別のシステム構成で使用する場合は，特に注意が必要です。たとえば，あるシステムで，ミニフロッピィディスクユニットだけが接続されている場合は，デバイス名「1：」はこのミニフロッピィディスクユニットのドライブ1になります。ところが，フロッピィディスクユニットとミニフロッピィディスクユニットが接続されている場合は，ミニフロッピィディスクユニットのドライブ1は「3：」となり「1：」は標準フロッピィディスクユニットのドライブ1になります。このような場合に備えて，プログラムの先頭で，dskf関数を使って接続されているフロッピィディスクユニットの種類を確認することをお勧めします。(dskf関数については，「BASIC REFERENCE MANUAL」を参照してください。)

12.2　プログラムのセーブ・ロード

12.2.1　プログラムのセーブ

フロッピィディスクにプログラムをセーブする場合は，カセットテープの場合と同じ様に，save文を使います。

save 〈ファイルディスクリプタ〉〔, |a / p|〕

例

```
save "prog 1"
save "prog 1 .n88"
save "2 : prog 1"
save "test 1", a
save "test 2", p
```

N88DISK-BASICでは，通常のプログラムのセーブ以外に，Aオプションと Pオプションを付けたセーブができます。このAオプションとPオプションについては，「12.2.3ファイル名のリストをとる」で説明します。

この他に，N$_{88}$DISK-BASICでは，機械語をフロッピィディスクにセーブすることもできます。この機械語のセーブには bsave文を使います。

bsave 〈ファイルディスクリプタ〉，〈開始番地〉，〈プログラムのバイト数〉

例

bsave "mprog" , &ha000, &h20

これは，A000H番地からA019H番地の機械語(長さ20H)を"mprog"というファイル名でフロッピィディスクにセーブする例です。

「12.1接続ディスクとデバイス名」のところでも述べましたがsave文およびbsave文では，デバイス名の指定に気をつけてください。ドライブ1にあるファイル名と同じファイル名でドライブ2にセーブしようとしてデバイス名を忘れると，ドライブ1のそのファイル名のプログラムが消えてしまいます。ドライブ2に"nec"というプログラムをセーブする場合

例

× **save "nec"**

○ **save "2 : nec"**

このことは，後述のファイル削除(kill)においても重要なことです。

Note **同じファイル名にセーブしようとした場合は，エラーにならずにそのファイル名のプログラムを消して上書きします。**

12.2.2 プログラムのロード

フロッピィディスクにセーブされているBASICのプログラムのロードは，load文によって行います。

load 〈ファイルディスクリプタ〉〔, r〕

例

load "prog 1"

load "prog . n88"

load "2 : prog 1"

load "prog 1", r

上の例の４番目は，Ｒオプション付きのload命令で，プログラムをロードした後で，そのプログラムを実行します。詳しくは，「12.2.8Ｒオプションとrun命令」を参照して下さい。

bsave文でセーブした機械語のロードには，bload文を使います。

bload 〈ファイルディスクリプタ〉〔,〔ロードアドレス〕〔, r〕〕

例

```
bload "mprog"
bload "mprog", &ha000
bload "mprog" , , r
```

１番目の例は，"mprog"というファイル名の機械語をbsave文でフロッピィディスクにセーブする際に指定した開始番地からロードします。２番目の例は，ロードする際，A000H番地から機械語をロードします。最後の例は，１番目の例と同様にプログラムをロードしたあとに，bsave文で指定した開始番地またはロードアドレスからプログラムをスタートさせます。(「12.2.8Ｒオプションとrun命令」を参照して下さい。)

データファイルをロードしたら？

このことは規定されていません。最悪の場合は，リセットをかけてもう一度始めから，N88DISK-BASICをスタートさせることになります。

この種のトラブルを避けるには，ファイル名でデータかプログラムを区別させるしかありません。その場合はファイルの拡張子をうまく利用して下さい。(「10.1ファイル」参照)

例

プログラムのファイルには必ず (".n88" をつける。 / 省略する。)

データファイルには". dat"をつける。

12.2.3　ファイル名のリストをとる

フロッピィディスクにセーブされているファイルの名前を知りたいときには，次の命令を使います。

files△〈デバイス番号〉

デバイス番号とは，ファイルディスクリプタのところで述べましたが，デバイス名での番号です。(省略するとデバイス番号は1です。)

例

files
progra. m　　　　**1**　(programというファイル名)
ok
files 2
ファイルなし
ok

上の例で，programのaとm(ファイル名の6文字目と7文字目)の間に小数点が表示されていますが，これは，programというファイルが非アスキー(バイナリー)形式でフロッピィディスクにセーブされていることを意味しています。

プログラムをフロッピィディスクにセーブするときは，非アスキー形式以外にアスキー(リスト)形式を指定することができます。

Note **アスキー形式とは，プログラムリストの1文字1文字を，そのままフロッピィディスクに書き込む形式です。**

Note **非アスキー形式というのは，プログラムリストの1文字1文字をそのまま書き込むのではなく，中間コードに変換し，プログラムを圧縮して書き込む形式です。**

これら2つの形式では，当然，非アスキー形式でセーブしたプログラムの方が容量が小さくてすみ，1枚のフロッピィディスクにより多くのプログラムをセーブすることができます。また，ロード・セーブに要する時間も短くなります。しかし，アスキー形式でセーブしておかなければできないこと(プログラムのマージとチェインです。12.2.4と12.2.9参照。)もあ

るので使いわける必要があります。

データファイルはすべてアスキー形式です。先程の例で，非アスキーセーブ(通常のセーブ)のファイル名の6文字目と7文字目の間に小数点が書かれていましたが，アスキー形式でセーブされたファイルの場合は，小数点のところが空白(スペース)になります。

アスキー形式でプログラムをセーブするときは，Aオプションを指定して次のようにします。

save 〈ファイルディスクリプタ〉, a

例

save "program1" , a	アスキー形式
save "program2"	非アスキー形式
files	
progra m1　1	アスキー形式
progra. m2　1	非アスキー形式

この例で，ファイル名の後に数字の1が出ていますが，これは，(「12.7 ディスクファイルの構造」で述べますが)クラスタと呼ばれるファイルの大きさを示す数です。

ファイル名の6文字目と7文字目の間の小数点または空白(スペース)は，files命令を実行したときに表示されるだけで，ロード(指定)するときは，そのファイル名を作ったときにつけたファイル名を用いることは，いうまでもありません。

例

先の例でセーブしたプログラムをロードするとき

load "program1"

load "program2"

(アスキー形式のファイル名をロードする場合でも，",a"は必要ありません。)

「12.2.1プログラムのセーブ」でふれたPオプションですが，これを指定すると，そのプログラムは暗号化された非アスキー形式でセーブされます。一度Pオプションを指定してセーブすると，そのプログラムの内容を見たり，変更したりできなくなります。このPオプションの指定を解除す

る方法は準備されていません。

Pオプションを指定してセーブしたプログラムは，その内容を見たり，変更することができません。

例

```
save "2：Program1", p
files
progra. m1            1     非アスキー形式
```

機械語のファイル(bsave文でセーブされたファイル)は，6文字目と7文字目の間に"＊"が表示されます。

```
bsave "2：mprog1", &ha000, &h20
files
mprog1 ＊             1     機械語のファイル
```

12. 2. 4　プログラムのマージ

メモリ上にあるプログラムと，指定されたフロッピィディスク上のファイル(アスキー形式で書かれたプログラム)を行単位(行番号ごと)で合成して，1つのプログラムにすることを「マージする」と言います。

マージする場合，同じ行番号のプログラムがあれば，フロッピィディスク上のファイルのその行番号のプログラムが優先して，メモリ上に置き替わります。マージできるファイルは「12.2.3ファイル名のリストをとる」(files命令)で述べたアスキー形式でセーブされたファイルだけです。マージした後，この合成されたプログラムをセーブしておけば，次回からは簡単にロードして使えるわけです。

merge 〈ファイルディスクリプタ〉

例

```
merge "program1"
```

program1はアスキー形式でセーブされたファイル。

マージできるプログラムはアスキー形式でセーブされたプログラムだけです。

12.2.5　ファイルの削除

フロッピィディスク上のファイルはkill命令で削除します。

kill 〈ファイルディスクリプタ〉

例

kill "prog1"

kill "2 : prog2. n88"

注　意

「12.1接続ディスクとデバイス名」のところでも触れましたが，デバイス名の指定には充分注意して下さい。一度killしたプログラムはもとにはもどりません。

12.2.6　ファイルの属性とset文

ファイルの属性には，次の2種が指定できます。

(1)　書き込み禁止

(2)　**read after write**（プログラムまたはデータのセーブを行った後，一度読み込んで検査します。）

通常，何も指定しなければ，どちらの属性も持っていません。特に重要なファイルであるならば，(1)の書き込み禁止属性を指定しておきましょう。ファイルの内容の変更が必要になった時には，その属性を解除して，ファイルの内容を変更してください。必要であれば，もう一度，書き込み禁止属性を指定しておきましょう。

属性の指定，解除にはset文を使います。

set 〈ファイルディスクリプタ〉，〈属性文字〉

属性文字はＰ，Ｒ，〈それ以外〉であり，それぞれ

Ｐ……書き込み禁止

Ｒ……read after write

それ以外(r ，p も含む)……属性の解除

例

set "prog1", "p"　書き込み禁止

set "prog1", "　"ファイル名prog1のすべての属性の解除

このファイルの属性は，フロッピィディスクに保存されます。

次に，他の形式のset文を紹介します。これは読みとばしてもかまいません。

(a) set 〈ドライブ番号〉，〈属性文字〉

現在そのドライブに入っているフロッピィディスクの属性を指定します。以後，このフロッピィディスクに作られるファイルは，その属性を持ちます。

例

set 1, "R" (read after write)

set 2, "P"(書き込み禁止)

(b) set ＃ 〈ファイル番号〉，〈属性文字〉

この文は，指定したファイル番号の属性を指定しますが，ファイルがオープンされている間だけ有効です。

例

set ＃1, "R" (read after write)

12.2.7 ファイル名の付けかえ

name命令によって，フロッピィディスクに書き込まれているファイル名を変更することができます。

name 〈旧ファイルディスクリプタ〉 as 〈新ファイルディスクリプタ〉

2つのファイルディスクリプタのデバイス名は，同じでなければなりません。

例

name "DEMO" as "SOFT"	○ファイル名"**DEMO**" →ファイル名"**SOFT**"
name "2：TEST" as "2：WORK"	○ファイル名"**TEST**" →ファイル名"**WORK**"
name "2：WORK" as "NEC"	× エラー

12.2.8　Rオプションとrun命令

メモリ上のBASICのプログラムを走らせるrun命令以外に，フロッピィディスク用のrun命令があります。

run 〈ファイルディスクリプタ〉

この形式のrun命令は，ファイルディスクリプタで指定されるプログラムをロードして，通常のrun命令を実行します。

例

run "test" → {load "test" / run}

Ｒオプションはload, bload, run命令に付加することができます。

load 〈ファイルディスクリプタ〉, r

bload 〈ファイルディスクリプタ〉, , r

run 〈ファイルディスクリプタ〉, r

loadのＲオプションとrunのＲオプションは，機能が全く同一で，指定したファイルのプログラムをロードしたのち，そのプログラムの実行をします。ここまでは

run 〈ファイルディスクリプタ〉

と同じですが，Ｒオプションが付くと，すべてのオープンされているデータファイルは，クローズせずに，オープンされたままになります。

この２つのＲオプション付き命令は，プログラムのチェイン(「12.2.9プログラムのチェイン」を参照)にも使えます。

bload文は，機械語プログラムでのＲオプションなので，データファイルのクローズ，オープンは関係ありません。ただロードした後，プログラムを実行するというだけのＲオプションです。

12.2.9 プログラムのチェイン

プログラムが大きくて，全部がメモリに入りきらない場合は，次のような方法が使われます。

(1) プログラムを幾つかのモジュールに分けます。

(2) それぞれ別のファイル名で，フロッピィディスクにセーブしておきます。

(3) 必要なモジュールのみをメモリにロードして実行します。

(4) その後，自動的に次のモジュールをロードして実行を続けます。

このような方法のことを“プログラムをチェインにする”と呼びます。

この方法は，12.2.8で述べたload, runのRオプションを使っても可能ですが，N88DISK-BASICでは，この方法にもっと適した命令を用意しております。それは，chain命令です。

chain 〔merge〕〈ファイルディスクリプタ〉
〔,〔行番号〕〔, all〕〔, delete〈範囲〉〕〕

chain命令は，次のような命令です。

(1) ファイルディスクリプタで指定したプログラムをロードして実行します。

(2) ロードする前にオープンされているファイルは，クローズされません。

(3) 行番号はロードしたプログラムの実行開始行を指定します。省略した場合はプログラムの最初から実行します。

(4) ロードする前の変数をそのまま使用することもできます。その場合は，allオプションを指定してすべての変数が使えるようにするか，common文を使って，必要な変数だけを使えるようにするかのどちらか一方の指定が必要です。(詳細は「BASIC REFERENCE MANUAL」のcommon文を参照して下さい。)

(5) mergeオプションは，ロードする前にメモリ上にあるプログラムと指定したプログラム(アスキー形式)をマージして，指定した行番号から実行させます。

(6) deleteオプションは，mergeオプション指定の時のみ付けられ，マージする前にメモリ上のプログラムの指定した範囲を削除します。

例えば，次のように指定します。

例

```
chain "work"
chain "work", 100, all
chain merge "program", 1000, all, delete 1000-2000
```

詳細は，「BASIC REFERENCE MANUAL」を参照して下さい。

12.3　データのロード・セーブ

12.3.1　シーケンシャルファイルとランダムアクセスファイル

N88DISK-BASICのデータファイルには，シーケンシャルファイルとランダムアクセスファイルがあります。

シーケンシャルファイルとは，連続的に書かれたデータファイルのことで，書いた順序でしか読めないファイルです。カセットテープに作られるファイルは，このシーケンシャルファイルです。

このファイルの特徴としては，次の事が挙げられます。

(1)　データのロード，セーブを行う場合は必ずファイルの先頭から順次行います。

(2)　ファイル内の一部のデータだけを書き変えることはできません。

(3)　ファイルの変更は，ファイルの最後に新たなデータを追加することだけです。

(4)　比較的簡単に使えます。

(5)　１つ１つのレコード(データ処理の単位で，レコードの集りをファイルと呼びます)の大きさは，自由に変えられます。

ファイル

レコード			レコード			
000001	ニチデンタロウ	ショムカ	000002	サイトウアキオ	パソコンカ	------
(社員番号)	(名　前)	(所　属)	(社員番号)			

これに対して，ランダムアクセスファイルはシーケンシャルファイルの不得意であるファイル内のデータ(レコード)を直接アクセスすることを得意とするファイルです。

このランダムアクセスファイルの特徴は,

(1) ファイル内の各レコードにランダムにアクセスできるので，データ(レコード)のロード，セーブを行う場合はファイルの先頭から，順次行われなくてもよい。

(2) ファイル内の一部のデータだけを書き変えることができます。

(3) ファイルの最後に，新たなデータを追加することもできます。

(4) シーケンシャルファイルに比べて，プログラムが複雑になります。

(5) 1つ1つのレコードの大きさは，256バイトに固定されています。

シーケンシャルファイルを使いこなせば，ほとんどのデータ処理ができますが，ランダムアクセスファイルを使いこなしてこそ，コンピュータのデータ処理機能をフルに活用したと言えるでしょう。

シーケンシャルファイルとランダムファイルで使用できる文と関数は次のとおりです。

<table>
<tr><th></th><th>シーケンシャルファイル</th><th>ランダムファイル</th></tr>
<tr><td rowspan="2">文</td><td colspan="2">OPEN
CLOSE</td></tr>
<tr><td>PRINT#
PRINT#　USING
INPUT#
LINE INPUT#
WRITE#</td><td>FIELD
LSET
RSET
PUT
GET</td></tr>
<tr><td>関数</td><td>INPUT$
EOF</td><td>LOC
LOF</td></tr>
</table>

12.3.2 シーケンシャルファイルにおける データのロード・セーブ

シーケンシャルファイルにおけるデータのロード・セーブはinput＃，print＃，write＃文およびそれらのバリエーションであるline input＃，print＃ using文で行います。

データのロード命令は，input＃，line input＃文で，セーブ命令はprint＃，write＃（print＃文と非常によく似ています。），print＃ using文です。（これらの命令の詳細は，「BASIC REFERENCE MANUAL」に載っています。）ここでは，シーケンシャルファイルにおけるデータのロード・セーブとデータの追加について，プログラム例をあげながら説明していきましょう。

次のプログラムをキーボードから入力してください。

プログラム	説明
100 open "DATA" as ＃1 ⏎	ファイルを開きます。
200 input "エイタンゴ"; a$ ⏎	
250 input "ヤク"; b$ ⏎	シーケンシャルファイルに書かせるデータを用意します（キーボードからデータを入力できるようにinput文を使用）。
300 write ＃1, a$, b$ ⏎	データをファイルに書き込みます（write＃文を使用）。
400 close ＃1 ⏎	データを書き込んだファイルから，データを読み出すために一度ファイルを閉じてから，再び開きます（「10.2 ファイルの OPEN, CLOSE」参照）。
450 open "DATA" as ＃1 ⏎	
500 input ＃1, x$, y$ ⏎	input＃文によりファイルから，データを読み出します。
550 print x$, y$ ⏎	画面に表示させます。
600 close ＃1 ⏎	ファイルを閉じます。
700 end ⏎	

入力は終わりましたか？

まず，プログラムリストを見てみましょう。次のキーを入力して下さい。

[f・4] [⏎]（またはlist[⏎]）

間違いはありませんね。もしあれば，入力し直して下さい。それでは実行してみましょう。

[f・5]（またはrun[⏎]）

すると画面に“エイタンゴ？”を表示してきました。何か英単語を入力してください。

例

```
run
エイタンゴ？　book [⏎]
ヤク　？　ホン [⏎]
book　ホン
ok
```

実行例をあげました。このような結果を得ることができましたか。

このプログラムでは1つの英単語しかファイルに書くことができません。次のようにプログラムを変えて下さい。

```
100 open "DATA" for append as #1 [⏎]
230 if a$="end" then 400 [⏎]
350 goto 200 [⏎]
570 goto 500 [⏎]
```

それでは実行してみましょう。

[f・5]（またはrun[⏎]）

ここでも，実行例を挙げておきましょう。

```
run
エイタンゴ　？　time [⏎]
ヤク　？　ジカン [⏎]
エイタンゴ　？　end [⏎]
book　ホン　time　ジカン
Input past end in 500
```

行番号100のopen文は，シーケンシャルファイルへデータを追加するた

めに，モードを変更するためのものです。(「10.2ファイルのOPEN, CLOSE」を参照。)こうしておけば先程のデータは消されることがありません。

行番号230でキー入力の終わりを検出します。

実行させたところ，input past end in 500というエラーメッセージが表示されました。これは行番号500から570までが，終わりのないループになっており，すべてのデータを読み終わっても，行番号500のinput#文を実行しようとしたために，表示されたわけです。このエラーをなくすためには，ファイルの終わりを検出して，その後input#文を実行しないようにすればよいわけです。ファイルの終わりを検出する関数としてeof(end of fileの略)関数があります。(「12.4関数」を参照して下さい。)

この関数を使って次のように変更してみましょう。

```
50 close
470 if eof(1) then 600
570 goto 470
```

行番号50のclose文は，エラーのためにクローズされなかったファイルを一度クローズするために入れておきます。

今後はどうでしょうか。実行させてみましょう。

[f・5] (またはrun [↵])

下の実行例のようになりましたか。

例

```
run
エイタンゴ？  desk
ヤク？        ツクエ
エイタンゴ？  end
book       ホン
time       ジカン
desk       ツクエ
ok
```

以上で，プログラムが完成しました。プログラムを実行させるごとに，単語数は増えていきます。

12.3.3 ランダムアクセスファイルにおけるデータのロード・セーブ

ランダムアクセスファイル(以後，ランダムファイルと呼びます)におけるデータのロード・セーブをデータバンクの例を用いて説明します。

(1) データバンクがあります。

(2) あなたは法律家です。

(3) そして，あなたはデータ・バンクに法律に関する資料を手に入れにきました。

(4) あなたはデータ・バンクの扉を開けます。

(5) 窓口がいくつかに分かれています。

(6) あなたは法律関係の窓口に行き，資料を手に入れます。

(7) そして，扉を閉めて事務所に戻りました。

それでは，この例を実際にランダムファイルを扱う場合に当てはめてみることにします。

データバンクはフロッピィディスクを意味しています。扉を開けるのはopen文で行います。窓口全体がランダムバッファで，ランダムファイルへのデータの出し入れを行う窓口です。ランダムバッファは1レコードで，

256バイトに固定されています。ランダムバッファは1つとは限らず，オープンするランダムファイルの数だけ持つことができます。(オープンできるファイルの数は「10.2ファイルのOPEN, CLOSE」と「10.3How many files?」を参照して下さい。)この窓口はいくつかに分かれています。これはfield文で指定します。この1つ1つの小さな窓口をフィールドと呼びます。資料(データ)はget文で手に入れ，close文で扉を閉じます。

以上は，データのロードの場合です。データのセーブの場合は，データバンク指定の書式(実際には, lest, rest文を用います)で書いた資料(データ)を法律関係の窓口に持って行き，保存してもらいます。(保存はput文で行います。)

ランダムファイルで扱うデータは文字でなければなりません。したがって，数値をセーブするときは，mki$, mks$, mkd$関数を使って文字に変換し，ロードするときは，cvi, cvs, cvd関数で数値に戻す必要があります。各関数については，「12.4関数」を参照して下さい。

具体的にはランダムファイルにおけるデータのロード・セーブは次のように行います。まずデータのセーブは次のような手順で行います。

(1) ファイルをオープンします。〔open文〕

```
open "DATA" as #1
```

ランダムファイルの場合，モードは指定しません。省いて下さい。(「10.2ファイルのOPEN, CLOSE」参照)

(2) ランダムバッファの中の各フィールドを指定します。〔field文〕

field #1, 50 as a$, 30 as b$, 80 as c$

この文の意味は，ファイル番号１用のランダムバッファ（256バイト）の最初の50バイトがa$，次の30バイトがb$，次の80バイトがc$の各文字変数に対応し，256－50－30－80＝96バイトは“あまり”になるということです。field文では，ランダムバッファをどのように割り付けるかの指定だけで，データをランダムバッファおよびフロッピィディスクに書くことは行いません。

(3)　各フィールドにデータをセットします。〔lest, rest文〕

各フィールドに対する文字変数にデータをセットします。セットする場合は，lest, rest文を使用しなければなりません。（対応する文字変数をlest, rest文の左辺，input文，read文等で代入できません。）

a$＝“abc”	×
input b$	×
read c$	×
lest a$＝“abcdef”	○（左づめ）
rest b$＝“Ok”	○（右づめ）
lest c$＝mki$（82）	○（左づめ）

（数値を文字に変換しています。）

この３つの文で，ファイル番号１のランダムバッファを次のように書き込みます。

ランダムバッファ＃１

abcdef	OK	82	
50バイト	30バイト	80バイト	96バイト あまり
a$ 左づめ	b$ 右づめ	c$ 左づめ	

データをセットしたフィールドで，データ以外の残りはスペースが入ります。逆に，lest, rest文でセットしようとした文字の長さがフィールドの長さよりも大きい場合，フィールドにはいりきらない文字は無視されます。

(4)　ランダムバッファの内容をディスクに書き込みます。〔put文〕

```
put ＃1, 2
```

この文によって，ファイル番号1のランダムバッファにセットされたデータをフロッピィディスクの“DATA”というファイル名のレコード番号2に書き込みます。フロッピィディスク上のランダムファイルには，レコード番号がついていて，通常はput文(get文も同様)ではこのレコード番号を指定します。しかし，このput文(get文も同様)のレコード番号は省略できます。省略した時はその直前にgetまたはputしたレコードの次のレコード番号が設定されます。

(5) ファイルを閉じます。〔close文〕

```
close ＃1
```

(1)〜(5)までの処理は，ランダムファイルにおけるデータのセーブです。データのロードは(3)と(4)が次のように変わります。

(3) ランダムファイル上の任意のレコードを読みます。〔get文〕

```
get ＃1, 2( 2 はレコード番号)
```

get文はディスクファイルから1レコード分のデータをランダムバッファに読み込みます。

(4) 各フィールドにセットされたデータを取り出して処理します。各フィールドにセットされたデータは，各フィールドに対応している文字変数で取り出すことができます。

```
print a$, b$, cvi(c$)
```

c$はputする際に数値を文字に変換したので，cvi関数を使って再び数値に変換する必要があります。

☆ランダムファイルにおけるデータのセーブプログラム例

```
10 open “DATA” as ＃1
20 field ＃1, 50 as a$, 30 as b$, 80 as c$
30 lset a$＝“abcdef”
40 rset b$＝“OK”
50 lset c$＝mki$(82)
60 put ＃1, 2
70 close ＃1
```

☆ランダムファイルにおけるデータのロードプログラム例

```
10 open “DATA” as ＃1
20 field ＃1, 50 as a$, 30 as b$, 80 as c$
30 get ＃1, 2
40 print a$, b$, cvi(c$)
50 close ＃1
```

12. 4　関　　数

ここではN88DISK-BASICの持つ関数について説明します。ディスク関数には次のようなものがあります。

eof	シーケンシャルファイルの終わりを検出
mki$	整数を2バイトの文字列へ変換
mks$	単精度実数を4バイトの文字列へ変換
mkd$	倍精度実数を8バイトの文字列へ変換
cvi	2バイトの文字列を整数型の数値に変換
cvs	4バイトの文字列を単精度実数型の数値に変換
cvd	8バイトの文字列を倍精度実数型の数値に変換
lof	ディスクファイルの大きさをセクタ数で表す
dskf	フロッピィディスクに関する情報を示す
input$	指定した長さの文字列をファイルから読み込む
loc	ランダムファイルの場合は最後に実行された時に使用されたレコード番号。シーケンシャルファイルの場合は今まで読み書きしたセクタ数
attr$	フロッピィディスクまたはファイルの属性
fpos	ファイルがある物理セクタ番号
dski$	指定したセクタから文字列を読み込む

これらのうちdski$については「12.5dski$とdsko$」で説明します。

(1)　**eof**

eofはシーケンシャルファイルが終わりかどうかを検出する関数で

す。ファイルからデータを繰り返して入力するような場合に，入力したデータが最後のデータかどうかを調べるのにeof関数を用います。

一般型は

eof(〈ファイル番号〉)
if eof (1) then goto 100

注　意

RS-232Cコミュニケーション・チャネル(COM：)をオープンしてデータを入力している場合については「BASIC REFERENCE MANUAL」の「EOF」の項を参照してください。カセットからデータを読み込んでいる場合は使用できません。(FCエラーとなります。)

(2) **mki$, mks$, mkd$**

ランダムファイルに数値データを書き込む場合，文字型に変換して書き込みます。その変換を行うのがmki$, mks$, mkd$です。

mki$は整数型の数値を2バイトの文字列に，mks$は単精度実数型の数値を4バイトの文字列に，またmkd$は倍精度実数型の数値を8バイトの文字列に変換します。

一般型は

mki$ (〈整数表記〉)
mks$ (〈単精度実数表記〉)
mkd$ (〈倍精度実数表記〉)

例

lset i$＝mki$(x%)

(3) **cvi, cvs, cvd**

数値型から文字型に変換されてファイルに書き込まれているデータを読み込んだ場合に，文字型からもとの数値型に変換する必要があります。

その変換を行うのがcvi, cvs, cvdです。

cviは2バイトの文字列を整数型の数値に，cvsは4バイトの文字列を単精度実数型の数値に，cvdは8バイトの文字列を倍精度実数型の数値にそれぞれ変換します。

一般型は

cvi（〈2バイトの文字列〉）

cvs（〈4バイトの文字列〉）

cvd（〈8バイトの文字列〉）

例

a＝cvs(x$)

(4)　**lof**

lofは，ディスクファイルの大きさをセクタ数で返す関数で，ディスクファイルがランダムファイルの場合はそれまでに書き込まれた最大のレコード番号を返します。ですからランダムファイルにおいて新しく書き込むデータのレコード番号を求めるような場合にlof関数を利用できます。

一般型は

lof(〈ファイル番号〉)

例

rec＝lof（1）＋1

(5)　**dskf**

dskfはディスクファイルの構造に関する情報を返す関数です。

一般型は

dskf(〈ドライブ番号〉〔,〈機能〉〕)

ここで〈機能〉は0～10の整数で指定し，それによってdskfは次のような機能を持っています。

0：片面あたりの最大トラック番号
1：1トラックあたりのセクタ数
2：片面の場合は0，両面の場合は1
3：1トラックあたりのクラスタ数
4：クラスタの総数
5：ディレクトリが格納されているトラック番号
6：1クラスタあたりのセクタ数
7：最初のFATが格納されているセクタ番号
8：最後のFATが格納されているセクタ番号

9：FATの個数

10：IDが格納されているセクタ番号

〈機能〉を省略した場合は，そのフロッピィディスクの残りの容量をクラスタ単位で返します。

例

crst=dskf(1，4)

(6) **input$**

input$は，シーケンシャルファイルから指定した長さの文字列を読み込んで返す関数です。

一般型は

input$(〈文字列の長さ〉〔,〈#ファイル番号〉〕)

です。

ファイル番号を省略した場合は，キーボードから入力した文字列を読み込みます。詳しくは「BASIC REFERENCE MANUAL」を参照してください。

例

x$=input$(5,#1)

この例では，#1のシーケンシャルファイルから文字数が5の文字列を読み込んでx$に代入します。

(7) **loc**

loc関数はランダムファイルとシーケンシャルファイルとで機能が異なります。

(i) ランダムファイルの場合

get文またはput文が最後に実行されたときに使用されたレコード番号を返します。

(ii) シーケンシャルファイルの場合

open命令を実行したときからの読み書きされたレコード数を返します。

一般型は

loc(〈ファイル番号〉)

例

j=loc(#1)

(8) **attr$**

attr$は，フロッピィディスクあるいはファイルの属性を返します。「属性」については12.2.6を参照して下さい。

一般型は次のとおりです。

(i) フロッピィディスクの属性

attr$(〈ドライブ番号〉)

(ドライブ番号：デバイス名で使っている数字)

(ii) ファイルの属性

attr$(〈#ファイル番号〉)

attr$("〈ファイル名〉")

それぞれ，指定したフロッピィディスク，オープンされているファイルの現在の属性，または，ファイル名で指定したファイルの固定属性(ディレクトリィに記録されている属性)を表す文字列を返します。

書き込み禁止	："△△P"
Read after write	："R△△"
属性なし	："△△△"

(△は空白文字)

例

print attr$("PROG. BAS")

この例では，ファイル"PROG. BAS"の属性を求めます。

(9) **fpos**

ファイル番号によって指定されたファイルが最後に読み書きした位置をセクタの通し番号で示します。

セクタの通し番号は次のように割当てます。

(i) 片面のフロッピィディスクの場合

トラック0のセクタ1に通し番号0を割当て，以下順に1切ざみで割当てていきます。例えば，トラック1のセクタ2は通し番号17になります。

(ii) 両面のフロッピィディスクの場合

サーフェス０のトラック０のセクタ１を通し番号０として，順にそのトラック内のセクタに番号を割当てます。

次に，サーフェス１，トラック０のセクタに番号を続けて割当てます。以下同様に，１トラックごとにサーフェスを替えて順に通し番号を割当てます。例えば，８インチ標準フロッピィディスクの場合は次のように割当てていきます。

サーフェス	0	0	0	……	0	0	1	1	1	……	1	1	0	0	0	……
トラック	0	0	0	……	0	0	0	0	0	……	0	0	1	1	1	……
セクタ	1	2	3	……	25	26	1	2	3	……	25	26	1	2	3	……
通し番号	0	1	2	……	24	25	26	27	28	……	50	51	52	53	54	……

一般型は

fpos(〈ファイル番号〉)

例

print fpos(1)

12. 5　dski$ と dsko$

ここでは，関数dski$とステートメントdsko$の説明をします。dsko$は非常に危険なステートメントです。したがって，このステートメントを使用するには十分注意をして下さい。このステートメントを実行する前に，使用するフロッピィディスクのバックアップを作っておくとよいでしょう。

(1)　**dski$**

フロッピィディスクの内容を読み込む場合，これまで説明してきたステートメントではファイル名かファイル番号でその内容を指定しましたが，dski$は物理アドレス(ドライブ番号，サーフェス番号，トラック番号，セクタ番号)で指定した内容を読み込む関数です。

(i)　片面のフロッピィディスクを使用する場合

一般型は

dski$(〈ドライブ番号〉，〈トラック番号〉，〈セクタ番号〉)

例

d$＝dski$(2 , 34, 16)

この例では，ドライブ 2 ，トラック34，セクタ16の 1 セクタの内容を読み込みます。

(ii)　両面のフロッピィディスクを使用する場合

一般型は

dski$(〈ドライブ番号〉,〈サーフェス番号〉,〈トラック番号〉,〈セクタ番号〉)

例

d$＝dski$(2, 1, 39, 16)

この例では，ドライブ 2 ，サーフェス 1 ，トラック39，セクタ16の 1 セクタの内容を読み込みます。

dski$は 1 セクタのデータ(256バイト)を文字列として返します。ところが文字変数の最大の長さは255文字なので，上の(i)，(ii)の例では 1 セクタの最後の1文字が失われます。

そこで256バイト全部を読み込む為に，dski$は＃ 0 のランダムバッ

ファに，指定したセクタの内容を読み込みます。

例

```
10 field #0, 128 as a$, 128 as a$
20 d$=dski$( 2 , 1 , 39, 16)
```

この例では，ドライブ2，サーフェス1，トラック39，セクタ16の1セクタの前半の128バイトをa$に，後半の128バイトをb$に読み込みます。またそのセクタの前から255バイトはd$に代入されます。

このようにdski$は，ランダムファイルと似たような使い方をします。

(2) **dsko$**

dsko$はdski$と同様に，#0のランダムバッファを使用します。まず，field #0によって指定したランダムバッファにデータを書き込みます。そしてdsko$によって，そのバッファの内容を指定したセクタに書き込みます。(field# 0の指定は「12.3.3」の中のfield文の説明を参照して下さい。)

(i) 片面のフロッピィディスクを使用する場合

一般型は

dsko$△ 〈ドライブ番号〉,〈トラック番号〉,〈セクタ番号〉

(ii) 両面のフロッピィディスクを使用する場合

一般型は

dsko$△ 〈ドライブ番号〉,〈サーフェス番号〉,〈トラック番号〉,〈セクタ番号〉

例

```
dsko$ 2, 1, 39, 16
```

この例では，ランダムファイルの場合と同様にしてセットした#0のランダムバッファの内容を，ドライブ2，サーフェス1，トラック39，セクタ16の1セクタに書き込みます。

12.6　フロッピィディスクの構造と取り扱い

12.6.1　各部の名称と機能

標識ラベル

標識ラベルには，そのフロッピィディスクの種類が書いてあります。たとえば両面用のフロッピィディスクには2D(2 sides, Double density)，片面用のフロッピィディスクには1D(1 side, Double density)などと書かれています。

このラベルは，出荷時に貼ってあるもので，ユーザがはがすことはできません。このラベルには，フロッピィディスクを使い始めてから廃棄するまでの間，不変の固定情報(たとえばフロッピィディスクの管理番号，使用開始年月日，持ち主の名前など)を書き込みます。

ジョブ用カラーラベル

このラベルはユーザが貼ったりはがしたりすることよできる一時貼着ラベルです。このラベルには，フロッピィディスクに記録されたデータ，またはその処理に関する情報などを記入します。

ライトプロテクトノッチ

このノッチにライトプロテクトシールを貼ることによって，機械的に書き込みを禁止させることができます。(標準フロッピィディスクの場合，通常は，機械的に書き込み禁止にすることはできません。)

ジャケット

フロッピィディスク本体を保護するための入れもので，内側には不織布が貼られています。

インデックス孔

フロッピィディスク本体とジャケットの両方にあいている孔で，セクタの位置を検出するためのものです。

ヘッドアクセス孔

ヘッド(レコードプレーヤーにたとえれば針にあたるもの)がフロッピィディスク本体に接するための孔です。

ライトプロテクトシール

ジャケットに切り込んであるライトプロテクトノッチに貼るシールです。(ミニフロッピィディスクで使います。)

保護封筒

フロッピィディスクを，ほこり，指紋，水分などから守る封筒です。ドライブに入れている時以外は，必ずこの封筒に入れておいてください。

12. 6. 2　フロッピィディスクの取り扱い

フロッピィディスク本体は，非常に薄い円板です(floppyは「ぺらぺらな」という意味です)。従って物理的に非常に弱く，取り扱いには，細心の注意が必要です。大切なデータを壊してしまわないために下記の注意を必ず守ってください。

フロッピィディスクの記録面を直接手でさわってはいけません。直接手でさわると，指紋などが付き，それが原因でゴミの付着を招き，フロッピィディスクやヘッドを損傷する原因にもなります。

直射日光や高温(65℃以上)にさらしてはいけません。

フロッピィディスクを曲げないでください。

クリップを使って，フロッピィディスクをはさんではいけません。

フロッピィディスクに磁石などを近づけてはいけません。50エルステッド以上の磁場にさらされると，記録されたデータがこわされるおそれがあり，また信号が乱されて，エラー発生の原因にもなります。

フロッピィディスクを保護封筒から取り出すには，フロッピィディスクの上端をつかんで引き出します。

フロッピィディスクの端部に力を加えたり，重いものをのせないでください。

ラベルへの記入方法

ジャケットに直接字を書かないでください。必ず所定のラベルを使用し，ラベルに書いてからジャケットに貼り付けてください。

古いラベルの記入欄がいっぱいになったら，古いラベルをはがし，きれいに接着剤をとった後に新しいラベルを貼ってください。古いラベルの上に，新しいラベルを重ねて貼ると，ラベルの厚みが増して，ドライブの障害の原因になるので，重ねて貼ることは避けてください。

カラーラベルは，フロッピィディスクに貼る前に，サインペン(フェルトペン)で記入しておきます。ボールペン，鉛筆，消しゴムの使用は，絶対に避けてください(屑や粉がジャケットの中に入ったり，ディスク面を傷つけたりして，故障の原因になります)。

フロッピィディスクに貼り付けてあるラベルに記入するときは，フロッピィディスクを必ず保護封筒に入れてからにします。フロッピィディスク自身をむきだしにした状態で，ラベルに記入すると，記録面を手で触れたりするおそれがあります。このとき，保護封筒の端部を押さえつけないでください。

平らな場所に，保護封筒入りフロッピィディスクを置いて記入してください。平らでないと，記入時にフロッピィディスクを変形させるおそれがあります。

ドライブへの装着方法

ディスクユニットに装着する時は，まっすぐにそう入し，止まる位置まで静かに押し込んでからドアを閉じてください。

斜めに押し込んだり，乱暴な装着は，フロッピィディスクの内径の縁を損傷して，フロッピィディスクを使用不能とすることがあるので注意してください。

フロッピィディスクを取り出した保護封筒は，ゴミなどの侵入を防止するため，収納箱に入れ，収納箱のふたをして，塵挨の少ない場所で保管してください。

保存方法

使用済みのフロッピィディスクは，必ず保護封筒に入れて収納箱の中で保管してください。収納箱の置き方は，水平，垂直を問いません。

使用してはいけないフロッピィディスク

次に述べるいずれかの項目に該当する場合は，そのフロッピィディスクの継続使用を禁止し，新しいフロッピィディスクに取り替える必要があります。

(1) フロッピィディスクが折れたり，ねじれたり，曲っていたり，一部が切れたり，端部がつぶれていたりなどの損傷を受けている場合。

(2) 清涼飲料，コーヒーなどのねばり気のある液体，溶剤や金属粉などで記録面やジャケットが汚れている場合。

(3) フロッピィディスクの記録面が損傷したり，汚れた場合。

(1)〜(3)のような場合，このままフロッピィディスクを装置へかけると，ヘッドやドライブを損傷したり，汚したりすることになり，ドライブの故障の原因になります。

また，ヘッドが汚れていたり，傷ついているのを知らないで，新しいフロッピィディスクをドライブへかけたりすると，このフロッピィディスクを汚したり，傷つけたりして，被害を広げることになります。

12. 7　ディスクファイルの構造

この節では，N$_{88}$DISK-BASICにおけるフロッピィディスクのファイル構造について説明します。この章は読みとばしてもかまいません。

12. 7. 1　フロッピィディスク

N$_{88}$DISK-BASICで使用できるフロッピィディスクは次の３種類です。

(1)　８インチ標準フロッピィディスク(両面倍密度用)

(2)　５インチミニフロッピィディスク(両面倍密度用)

(3)　５インチミニフロッピィディスク(片面倍密度用)

フロッピィディスクにはメモリと同じように番地がついています。フロッピィディスクの番地指定にはサーフェス番号(両面倍密度のときのみ必要。０か１。)，トラック番号，セクタ番号を使用します。図12.7.1は８インチ両面倍密度，図12.7.2は５インチ両面倍密度，図12.7.3は５インチ片面倍密度のフロッピィディスクの概略です。

図12.7.1　８インチ両面倍密度フロッピィディスク

セクタ16 セクタ1
セクタ15 セクタ2
セクタ3
トラック0
トラック1
トラック38
トラック39

図12.7.2　5インチ両面倍密度フロッピィディスク

セクタ16 セクタ1
セクタ15 セクタ2
セクタ3
トラック0
トラック1
トラック33
トラック34

図12.7.3　5インチ片面倍密度フロッピィディスク

図をみてもわかるように，トラック数，セクタ数はフロッピィディスクの種類によって異なります。これを表にまとめると次のようになります。

	サーフェス番号	トラック数	トラック番号	セクタ数	セクタ番号
8インチ両面倍密度	0か1	片面77(154)	0～76	26	1～26
5インチ両面倍密度	0か1	片面40(80)	0～39	16	1～16
5インチ片面倍密度	必要なし	35	0～34	16	1～16

Note

サーフェス番号0はおもて面，サーフェス番号1はうら面です。3種類とも，1セクタは256バイトからなります。フロッピィディスクの場合，最少読み書き単位はセクタです。

12.7.2 クラスタ

フロッピィディスクの読み書きの最少単位は1セクタですが，N88DISK-BASICは26セクタ(5インチミニフロッピィディスクの場合は8セクタ)を1まとめにして，これを管理の最少単位としています。これを〈クラスタ〉と呼びます。クラスタとサーフェス，トラック，セクタの関係は次の表のようになっています。

8インチ両面倍密度フロッピィディスクの場合

クラスタ	サーフェス	トラック	セクタ
0	0	0	1～26
1	1	0	1～26
2	0	1	1～26
3	1	1	1～26
4	0	2	1～26
⋮	⋮	⋮	⋮
152	0	76	1～26
153	1	76	1～26

5インチ両面倍密度フロッピィディスクの場合

クラスタ	サーフェス	トラック	セクタ
0	0	0	1～8
1	0	0	9～16
2	1	0	1～8
3	1	0	9～16
4	0	1	1～8
⋮	⋮	⋮	⋮
158	1	39	1～8
159	1	39	9～16

5インチ片面倍密度フロッピィディスクの場合

クラスタ	トラック	セクタ
0	0	1～8
1	0	9～16
2	1	1～8
3	1	9～16
4	2	1～8
⋮	⋮	⋮
68	34	1～8
69	34	9～16

12.7.3 トラックの割りあて

(1) 8インチ両面倍密度フロッピィディスク

(a) システムディスク（N88DISK-BASICを含むディスク）

サーフェス	トラック	セクタ	内容
0	0	すべて	未使用
	1	1	IPL
	1	2～26	DISK code
	2	すべて	DISK code
	3～34	すべて	ユーザ領域
	35	1～22	ディレクトリ
	35	23	ID
	35	24～26	FAT
	36～76	すべて	ユーザ領域
1	0	すべて	未使用
	1	すべて	DISK code
	2～76	すべて	ユーザ領域

(b)　データディスク(N88DISK-BASICを含まないディスク)

サーフェス	トラック	セクタ	内　　容
0	0	すべて	未使用
	1～34	すべて	ユーザ領域
	35	すべて	システムディスクと同じ
	36～76	すべて	ユーザ領域
1	0	すべて	未使用
	1～76	すべて	ユーザ領域

(2)　5インチ両面倍密度フロッピィディスク

(a)　システムディスク(N88DISK-BASICを含むディスク)

サーフェス	トラック	セクタ	内　　容
0	0	1	IPL
	0	2～16	DISK code
	1	すべて	DISK code
	2～39	すべて	ユーザ領域
1	0～1	すべて	DISK code
	2～17	すべて	ユーザ領域
	18	1～12	ディレクトリ
	18	13	ID sector
	18	14～16	FAT
	19～39	すべて	ユーザ領域

(b) データディスク(N88DISK-BASICを含まないディスク)

サーフェス	トラック	セクタ	内　　容
0	0～39	すべて	ユーザ領域
1	0～17	すべて	ユーザ領域
	18	すべて	システムディスクと同じ
	19～39	すべて	ユーザ領域

(3) 5インチ片面倍密度フロッピィディスク

(a) システムディスク(N88DISK-BASICを含むディスク)

トラック	セクタ	内　　容
0	1	IPL
0	2～16	DISK code
1～3	すべて	DISK code
4～17	すべて	ユーザ領域
18	1～12	ディレクトリ
18	13	ID sector
18	14～16	FAT
19～34	すべて	ユーザ領域

(b) データディスク(N88DISK-BASICを含まないディスク)

トラック	セクタ	内　　容
0～17	すべて	ユーザ領域
18	すべて	システムディスクと同じ
19～34	すべて	ユーザ領域

12.7.4 ディレクトリ，FAT

N$_{88}$DISK-BASICで，ディスクファイルを扱うときはすべてファイル名で行います。したがって，サーフェス番号，トラック番号，セクタ番号を使う必要はほとんどありません。ファイル名を，実際のファイルのサーフェス番号，トラック番号，セクタ番号に変換する仕事はN$_{88}$DISK-BASICが行っています。

ユーザ
ファイル名
N$_{88}$DISK-BASIC
サーフェス番号
トラック番号 セクタ番号
フロッピィディスク
FAT
ディレクトリ

ファイル名をフロッピィディスク上のアドレス(サーフェス番号，トラック番号，セクタ番号)に変換するために，N$_{88}$DISK-BASICでは2つの表を管理しています。〈ディレクトリ〉と〈FAT〉がそれです。

ディレクトリは，そのディスクに格納されているすべてのファイルの名前，属性，格納されている場所(の先頭)の表です。

各ディレクトリは16バイトからなります。各バイトは次のような内容を持っています。

バイト	内　容
0～5	ファイル名
6～8	拡張子
9	属性
10	そのファイルが格納されている先頭のクラスタ番号
11～15	未使用

属性は次のようになっています。

値(16進数)	意　　味
0	ASCII形式
1	機械語ファイル
10	ASCII形式，書き込み禁止
40	ASCII形式，リードアフターライト
80	非アスキー形式
90	非アスキー形式，書き込み禁止
A0	非アスキー形式，暗号化
C0	非アスキー形式，リードアフターライト

ファイル名の先頭バイトの内容がFF(16進数)のとき，そのディレクトリは未使用であることを示します。0のとき，削除(kill)されたファイルであることを示します。

FATは各クラスタの使用状態を示す表です。12.7.3のトラックの割りあての表をみてもらえばわかるように，FATは3セクタにわたって書き込まれています。各セクタは全く同じ内容です。フロッピィディスクをアクセスしたときに，FATの内容(3セクタ)がメモリにコピーされます。それぞれのセクタの0～153(8インチ両面倍密度の場合)バイトがクラスタ0～153にそれぞれ対応しています。すなわち各バイトの持っている値が，各クラスタの使用状態を示しているのです。

バイトのデータ　(16進)	クラスタの使用状態
8インチ両面の時　0～99 5インチ両面の時　0～9F 5インチ片面の時　0～45	使用中。連続したクラスタの一部であり，後続するクラスタを持つ。値が，後続するクラスタの番号を示している。
8インチ両面の時　C1～DA 5インチの時　C1～C8	使用中。連続したクラスタの最後のクラスタであり，下5(5インチの時は下4)ビットの内容が，そのクラスタで実際に使われているセクタの数を表わす。
FE	予約ずみのクラスタで，ファイルとして使うことはできない。(DISKコード，IPL，ディレクトリ，FAT自身を含むクラスタがこれである。)
FF	未使用

なお，表の説明で「連続したクラスタ」といいましたが，これは物理的に連続したクラスタを意味しません。N88DISK-BASICが管理するファイルは一般に物理的(地理的)にとびとびのクラスタを結いで構成されているのです。

12.7.5 ID

ID部には１セクタ(＝256バイト)が割り当てられています。各バイトは次の様な意味を持っています。

属性	ファイル数	BASICテキスト
1バイト	1バイト	254バイト

最初の１バイトは，そのディスクの属性を示しています。値とその意味はディレクトリ部のものと同じです。

通常のシステムディスクでは〈ファイル数〉として255(16進数ではFF)が指定されています。この場合，システム起動時にキーボードからファイル数を入力する必要があります。このファイル数に0～15を指定すると，N88DISK-BASICをスタートさせる時に

How many files(0-15)?

を表示せずに，自動的にそのファイル数が設定されます。

ファイル数の設定が終わると(ファイル数を設定せずに，「How many files?」に対してキーボードから入力を行った場合でも)，次のBASICテキストの内容が自動的に実行されます。つまり，N88DISK-BASICをスタートさせる場合，システムディスクのIDに書かれているBASICテキストの内容は常に実行されます。

また，ディスクユーティリティ「SET UP INFO」により，年の設定を行った場合にもこのBASICテキストのエリアが使われます。

IDは，ファイル数が255(FFH)，BASICテキストがすべて０か32(０か20H)が通常の状態です。

例

IDセクタ

00 05 72 75 6E 22 41 00 …………00

(r)(u)(n)(")(A)(16進数)

この例の様にIDセクタの内容が書かれている場合，自動的に5個のファイルをアロケートしてファイルAのプログラムが走り出します。

IDの書き替えを行いたい場合は，システムディスクに入っているディスクユーティリティ(dskut2)の「SET UP INFO」を使って下さい。

第13章

第13章
機械語モニタ

この章は，PC-8801MKⅡで直接機械語を扱う場合に使用する機械語モニタについて説明します。

13.1　概　　要

PC-8801MKⅡには，独立したI/Oコントロール・ルーチンを持つ強力な機械語モニタが準備されています。この機械語モニタの主な特長は次のとおりです。

1. ミニアセンブラ機能を持っています。
2. 数値を16進数，8進数のどちらでも入力，表示が可能です。
3. I/Oポートへの入出力が可能です。
4. 逆アセンブル機能を持っています。
5. CPUのレジスタの値の表示，変更が可能です。
6. カセットだけではなく，フロッピィディスクへもロード，セーブが可能です。
7. スクリーンエディタによるメモリの内容の変更ができます。
8. Helpコマンドにより，コマンドの入力形式がわかります。

13.2 機械語モニタの動かし方

機械語モニタが機能する環境は，１つの実行モードとして，BASICモードとは別な実行環境をもっています。これを“機械語モニタモード”とよびます。ターミナルモードと同様に，BASICモードから切り替えて使用します。

(1) 実行モードとしての遷移図は，次のようになります。

BASICモード → mon ⏎ → 機械語モニタモード

機械語モニタモード → CTRL + B → BASICモード

(2) BASICモードから機械語モニタモードへの切り替え方法

画面	説明
OK ■	BASICコマンド入力可能状態です。
OK mon ⏎	monコマンドを入力します。
OK mon h〕■	monコマンドの実行完了により，機械語モニタモードになります。 h〕が表示されます。

h〕によって，機械語モニタモードになっていることを示します。

N-BASIC(DISK-BASIC)モードの場合は，h〕の代りに，＊が促進メッセージとして表示されます。

(3) 機械語モニタモードからBASICモードへもどるための方法

画面	説明
h〕■	機械語モニタ用コマンド入力可能状態です。
h〕CTRL + B	CTRL キーと B キーとを同時に押下します。
h〕∧B OK ■	OKが表示され，次の行にカーソルが表示されます。 BASICモードになったことを示します。

13. 3　機械語プログラムのメモリ配置

N88(DISK)-BASICを使っているときのメモリマップは，次のようになっています。

アドレス	内容
FFFF	N88-BASICワークエリア
E600	機械語エリア
YYYY	変数エリアなど　←CLEAR , YYYYで指定
XXXX	(DISK code) ファイルバッファなど
8400	テキストウィンドウ
8000	N88-BASICインタプリタ ROM
0000	

機械語のプログラムを作る場合は，CLEAR命令で，機械語エリアを確保しておかなければなりません。

CLEAR, YYYY

とすることで，図の斜線の部分を機械語エリアとして使うことができるようになります。

それ以外の部分は，N88(DISK)-BASICが使っていますので，内容を変更するとBASICモードに戻ったときの動作が保障されません。

13. 4　コマンドの種類

機械語モニタのコマンドには，表に示すように22種類のコマンドがあります。(命令は大文字でも小文字でも入力できます。)

コマンド	意　味	機　能
A	アセンブル	入力した1行をアセンブルします。
B	ベース	数値の表現形式を変えます。(8進↔16進)
D	ダンプ・メモリ	メモリの内容をディスプレイに表示します。
E	エディト・メモリ	スクリーン・エディタの機能を用いてメモリの内容を変更します。
F	フィル・メモリ	メモリの内容を定数で埋めていきます。
G	ゴー	ユーザープログラムを実行します。
I	インプット	I/Oポートの値を読み込みます。
L	ディス・アセンブル	機械語を逆アセンブルします。
M	ムーブ・メモリ	ある範囲のメモリの内容を他のアドレスのメモリ領域へ移します。
O	アウトプット	I/Oポートへデータを出力します。
P	プリンタ・スイッチ	プリンタへの出力をコントロールします。
R	リード・テープ	カセットテープからデータをロードします。
S	セット・メモリ	メモリにデータをセットします。
TM	テスト・メモリ	メモリをテストします。
V	ベリファイ・テープ	カセットテープの内容と，メモリの内容を比較します。
W	ライト・テープ	メモリの内容をカセットテープにセーブします。
X	イグザミン・レジスタ	CPUのレジスタの値を調べ,変更します。
HELP または CTRL-A	ヘルプ	コマンドとそのパラメータの形式をディスプレイに表示します。

コマンド	意　　味	機　　能
CTRL-B	リターン	BASICモードへ復帰します。
CTRL-D*	ダンプ・ディスク	フロッピィディスクの内容をディスプレイに表示します。
CTRL-R*	リード・ディスク	フロッピィディスクから，データをロードします。
CTRL-W*	ライト・ディスク	フロッピィディスクへデータをセーブします。

*がついている3つのコマンドは，N88DISK-BASICでのみ機能します。

13.5　コマンド使用上の規則

(1)　数値の入力

機械語モニタで扱う数値形式には，16進数モードと８進数モードがあります。この切り替えは，Ｂコマンドで行います。システム既定値は16進数モードです。

①　16進数モード

アドレス　　：４桁　０～FFFF

メモリの内容：２桁　０～FF

②　８進数モード

アドレス　　：６桁　０～177777

メモリの内容：３桁　０～377

アドレスの場合は，入力された数値の終わりから，16ビットが有効になります。メモリの内容の場合は，終わりから，８ビットが有効になります。

数値の演算機能があります。

各コマンドで，アドレスやメモリの内容を入力する場合，加減の演算機能が使用できます。（ただし，括弧や累乗は使えません。）

例

h〕s9000＋100 ⏎

9100 00 —

(2)　カセットテープのスピード

機械語モニタでは，Ｒコマンド，Ｖコマンド，Ｗコマンドで，カセットテープを使用します。このとき，ファイル名の指定の方法により，転送速度が決まります。

一般形式

r〔1: 2:〕〔〈ファイル名〉〕

w〔1: 2:〕〔〈ファイル名〉〕，〈セーブ開始アドレス〉，〈セーブ終了アドレス〉

v〔1: 2:〕〔〈ファイル名〉〕

ファイル名	転送速度
[〈ファイル名〉] 1: [〈ファイル名〉]	1200ボー
2: [〈ファイル名〉]	600ボー

[〈ファイル名〉]は省略可

(3) 機械語モニタモードのディスプレイ画面形式

① BASICモードからの切り替え直前のWIDTH文による状態がひきつがれます。

② Eコマンドは，テキスト画面のスクロールウィンドウによっては，動作しない場合があります。

機械語モニタモードに入る前に，

CONSOLE 0, 〈17以上の値〉

を実行しておいてください。

13. 6　コマンドの説明

(1) A(アセンブル)

入力形式：**a** 〔**〈格納開始番地〉**〕

Intel 8080ニーモニックで入力した1行のテキストをアセンブルし，できた機械語をメモリに格納します。そのために，Aコマンドでは，普通は機械語の格納を開始する番地を指定します。(格納開始番地の指定を省略した場合は，その前に実行したAコマンドの実行終了番地の次の番地となります。)

h〕 a9000 [⏎]

Aコマンドを入力すると，格納開始番地を表示して，ニーモニックの入力待ちになります。そこで，Intel 8080ニーモニック命令を入力し，リターンキー([⏎])を入力すると，その一行をアセンブルし，できた機械語をアドレスのすぐ横に表示し，そのアドレスに格納します。そして次の行に，次に機械語を格納する番地を表示し，ニーモニック命令の入力待ちになります。

```
h〕 a9000 ⏎
9000 3E 01 mvi a, 1 ⏎
9002
```

入力したニーモニック命令にエラーがあった場合は，"?"を表示してモニタのコマンド入力待ちになります。このときは，もう一度アドレスを指定しなくても，コマンドだけ入力すれば，エラーを起こしたときのアドレスから実行を開始します。

```
h〕 a9000
9000 3E 01 mvi a, 1 ⏎
9002        mvl b, 1 ⏎
?
h〕 a ⏎
9002
```

このアセンブラは，すべてリターンキー(⏎)が入力されるまで，読み込みを続けます。また，INS DEL キーが使えますから，まちがえて入力した文字を訂正できます。このアセンブラから抜け出すには，アドレスが表示されたあと，リターンキー(⏎)を入力するか，STOP キーもしくは，**CTRL-C**を入力します。

このアセンブラで使用できるニーモニックは，Intel 8080ニーモニック命令とZ80ニーモニック命令の中のレラティブジャンプ命令です。Z80のレラティブジャンプ命令の場合は，絶対アドレスを入力すれば，アセンブラが相対アドレスに変換してくれます。

Z80のレラティブジャンプ命令の入力形式は，次のとおりです。

Z80	モニタで入力するとき
jr 〈アドレス〉	jmpr 〈アドレス〉
jr z, 〈アドレス〉	jrz 〈アドレス〉
jr nz, 〈アドレス〉	jrnz 〈アドレス〉
jr c, 〈アドレス〉	jrc 〈アドレス〉
jr nc, 〈アドレス〉	jrnc 〈アドレス〉
djnz 〈アドレス〉	djnz 〈アドレス〉

参　照

Intel 8080ニーモニック命令についてはμCOM80ユーザーズマニュアルを参照してください。

(2)　B(ベース)

数値を取り扱う形式を設定するコマンドです。数値をキーボードから入力したり，ディスプレイに表示する場合に，16進数として扱うか，8進数として扱うかを決めます。N88-BASICモードから，モニタへ切り替えたときは，16進数モードになっていて，コマンド入力待ちになると，

h〕

が表示されます。8進モードに切り替えるには，"bq"と入力します。するとq〕が表示され，8進数モードになったことがすぐにわかります。

h〕 bq ⏎
q〕

再び16進数モードへ切り替えるには，"bh"を入力します。

q〕 bh ⏎
h〕

(3)　D(ダンプ・メモリ)

メモリの内容をディスプレイに表示するコマンドです。16進モードになっている場合は，その数に対応するアスキー文字も表示されます。さらに，プリンタ・スイッチでプリントモードが指定されていれば，結果がプリンタへも出力されます。(プリンタ・スイッチおよびプリントモードについては，(11)P(プリンタ・スイッチ)を参照してください。)

Dコマンドは次の形で入力されます。

d〔<表示開始アドレス>〕〔，<表示終了アドレス>〕

ですから，実際の入力は，次の形になります。

(i)　**d9000, 9030** ⏎
(ii)　**d9000** ⏎
(iii)　**d, 9030** ⏎
(iv)　**d** ⏎

(i)の場合が標準で，9000番地から9030番地までのメモリの内容が表示されます。(ii)のように〈表示開始アドレス〉だけ指定した場合は，〈表示開始アドレス〉から，16バイト分のメモリの内容が表示されます。逆に(iii)のように〈表示終了アドレス〉だけ指定した場合は，その前に実行されたDコマンドの最後のアドレスの次のアドレスから，〈表示終了アドレス〉までのメモリの内容を表示します。(iv)のように〈表示開始アドレス〉も〈表示終了アドレス〉も指定しない場合は，このコマンドが入力される前に実行されたDコマンドの最後のアドレスの次のアドレスから16バイト分のメモリの内容が表示されます。

Dコマンドを実行中に，その実行を中止したい場合は，STOP キーもしくは，CTRL-Cを入力します。画面にメモリの内容を表示している場合，途中で表示を一時停止させるには，CTRL-Sを入力します。再び表示を再開させる場合には，STOP キーとCTRL-C以外を入力します。

また，Dコマンド終了後にリターンを入力すると(iv)と同じ動作を行います。

(4)　E(エディト・メモリ)

Eコマンドは，スクリーンエディタの機能を十分に活用して，スクリーンに表示されたメモリの内容を変更するコマンドです。Eコマンドは，次の形式で入力します。

e〔〈開始アドレス〉〕

〈開始アドレス〉を省略した場合は，最後にEコマンドで内容を変更したメモリのアドレスが，〈開始アドレス〉となります。

Eコマンドを実行して，メモリの内容をディスプレイに表示した後は，カーソル移動キー(↑ ↓ ← →)とロール・アップ，ロール・ダウンキー(ROLL UP ROLL DOWN)を使って変更したいメモリの内容のところへカーソルを移動させ，新しい数値を入力します。画面上で変更された数値はそのまま，メモリの内容も変更されています。エディトメモリを終了するときはSTOPまたは ESC を入力します。

(5)　F(ファイルメモリ)

アドレスで指定した範囲のメモリの内容を，指定した定数の値に置き替える命令です。コマンドの形式は次のようになっています。

f 〈開始アドレス〉,〈終了アドレス〉,〈定数〉

Fコマンドでは，各パラメータを省略することはできません。コマンドが正常に実行されたかどうかは，Dコマンドを用いて確認します。

```
h〕 f9000, 9010, 0 ⏎
h〕 d9000, 9010 ⏎
9000 00 00 00 00 00 00 00 00
9008 00 00 00 00 00 00 00 00
9010 00
h〕
```

(6)　G(ゴー)

指定したアドレスにジャンプして，そこからプログラムの実行を開始します。このとき，２つまで〈ブレイク・ポイント・アドレス〉を指定できます。〈ブレイク・ポイント・アドレス〉とは，プログラムの実行中，実行アドレスが〈ブレイク・ポイント・アドレス〉となったとき，実行を中止して，コマンドの入力待ち状態にさせるアドレスです。

g〔〈実行開始アドレス〉〕,〔,〈ブレイク・ポイント・アドレス＃1〉〔,〈ブレイク・ポイント・アドレス＃2〉〕〕

〈実行開始アドレス〉を省略した場合，最後に実行した〈ブレイク・ポイント・アドレス〉が〈実行開始アドレス〉になります。〈ブレイク・ポイント・アドレス〉で実行が中断された場合は，CPUのレジスタの値は保存されます。

注　意

ブレイク・ポイント・アドレスは，指定したアドレスが，RAM領域にあるときだけ有効になります。

(7)　I(インプット)

入力ポートの値を読み，画面に表示します。〈ポートアドレス〉を入力した後，スペースを入力すれば，その行に入力ポートの値を表示し，リターンキー(⏎)を入力した場合は，次の行に入力ポートの値が表示されます。

i 〈ポート・アドレス〉⏎（または，スペースキー）

（Oコマンドの項を参照にしてください。）

(8)　L（ディス・アセンブル）

メモリ・イメージをIntel 8080ニーモニックに逆アセンブルして表示するコマンドです。さらに，Z80のレラティブ・ジャンプ命令も使用できますが，アドレスは，絶対アドレスで表示されます。Lコマンドの形式は次のようになっています。

l〔〈逆アセンブル開始アドレス〉〕〔，〈逆アセンブル終了アドレス〉〕

〈逆アセンブル開始アドレス〉が省略された場合は，最後に逆アセンブルされたアドレスの次のアドレスが，〈逆アセンブル開始アドレス〉になります。また，〈逆アセンブル終了アドレス〉が省略された場合は，〈逆アセンブル開始アドレス〉＋15が，〈逆アセンブル終了アドレス〉となります。

Lコマンドの実行中に，実行を中断したい場合は，CTRL-Cまたは，**STOP**キーを入力します。実行を一時中断したい場合は，CTRL-Sを入力します。実行を再開するには，**STOP** キーおよびCTRL-C以外の任意のキーを入力します。また，Lコマンドで表示されるニーモニックのオペランドは，現在，指定されている数値モードで表示されます。

さらにLコマンド終了後にリターンキーを入力すると，〈逆アセンブル開始アドレス〉と〈逆アセンブル終了アドレス〉を省略したときと同じように〈逆アセンブル終了アドレス〉から16バイト分のメモリ・イメージを逆アセンブルします。

(9)　M（ムーブ・メモリ）

Mコマンドは，メモリ内で，ある位置から別の位置へ，メモリ領域のブロック転送を行います。Mコマンドは，次の形式で入力します。

m〈転送するメモリ領域の先頭番地〉，〈転送するメモリ領域の最終番地〉，〈転送先の先頭番地〉

たとえば，9000番地から9100番地のメモリ内容をA000番地からA100番地へ転送する場合は，

h〕m9000, 9100, a000 ⏎

と入力します。この後，Dコマンドを実行すれば，転送できたことを確認できます。

(10) O(アウトプット)

このコマンドは，指定したポートアドレスに，データを出力します。Oコマンドの入力形式は，

o 〈ポートアドレス〉，〈データ〉

となります。

Iコマンドおよび，Oコマンドは，自作のボードを機械語プログラムでコントロールする場合など，特殊な用途に使用するコマンドです。

(11) P(プリンタ・スイッチ)

Dコマンド，Lコマンド，CTRL-Dコマンドを実行した結果を，プリンタに出力するかどうかを決定します。

モニタに入ったときは，プリンタ・スイッチは「OFF」になっていて，Dコマンド，Lコマンド，CTRL-Dコマンドを実行しても，結果はディスプレイに表示されるだけです。このときPコマンドを入力すると，プリンタ・スイッチは「ON」になり，以後，実行されるDコマンド，Lコマンド，CTRL-Dコマンドの結果は，プリンタにも出力されます。

プリンタ・スイッチが「ON」になると記号がh)，q)，になります。

Note

プリンタ・スイッチ 「ON」 h)，q) ← p⏎ → **プリンタ・スイッチ 「OFF」h]，q]**

注　意

モニタを使用しているときも，**COPY** キーは使用できます。ですから，画面のハードコピーを取る場合は，**COPY** キーを使用してください。

(12) R(リード・テープ)

カセットテープからデータをロードするコマンドです。Rコマンドは，次の形式で入力します。

r〔〈ファイル名〉〕

ファイル名を指定した場合は，カセットテープのデータのファイル名をチェックし，指定したファイル名が見つかると，そのデータをメモリにロードします。このとき，指定したファイル名以外のファイル名が見つかった場合は，

Skip　　XXXX

とディスプレイに表示し，指定したファイル名が見つかるまで読み続けます。指定したファイル名が見つかった場合は，

Found　　XXXX

とディスプレイに表示し，その下にロードしているデータを表示します。

ファイル名を指定しない場合は，最初に見つかったファイルをロードします。

ファイル名は，6文字以下です。6文字を越えた場合は，最初の6文字が有効になります。

(13)　S(セット・メモリ)

Sコマンドは，メモリの内容を確かめたり，変更するコマンドです。Sコマンドは次の形式で入力します。

s〔〈開始アドレス〉〕 [⏎]（または，スペースキー）

Sコマンドで，〈開始アドレス〉を指定した後，リターンキー([⏎])を入力すると，次の行にアドレスとその内容を表示し，数値の入力待ちになります。スペースキーを入力すると，改行せずに，メモリの内容を表示し，数値入力待ちになります。

入力する数値は，16進モードならば，2桁の16進数を入力しますが，2桁以上の数値を入力した場合は，最後の2桁が有効となります。数値を入力した後，スペースキーを入力すれば，メモリに値をセットし，次のアドレスのメモリの内容を表示します。また，数値を入力した後，リターンキーを入力すれば，新しい値がセットされ，コマンド入力待ちになります。もし，メモリの内容を変更しないならば，数値を入力せずにスペースキーを入力すれば，次のアドレスに移ります。アドレスを1つ前に戻したい場合は，CTRL-Bまたは[←]を入力します。

(14)　TM(テストメモリ)

PC-8801MKⅡに実装されている，ユーザーズメモリ64Kバイトと，グラフィック用VRAM 48Kバイトをテストするコマンドです。テストは，次の順序で行います。

①　モニタに入る前に“screen△0,2 [⏎]”を入力する。

② モニタへ入る。

mon ⏎

③ tm ⏎

④ 次の順でチェックします。

テキストVRAM

変数・ワークエリア

グラフィック用VRAM

テキストエリア

このテストには，約12分間かかります。テストが無事に終了すると，

test complete!

というメッセージが現われますから，リセットボタンを押して下さい。

エラーが起きた場合，(1)で起きたときは，ブザー(beep音)が鳴り続けます。(2)以後の場合は，エラーが起きた番地を表示します。

Note **TMコマンドでエラーが発生した場合は，もよりのBit-INNもしくは，お買い上げの販売店で御相談ください。**

(15) V(ベリファイ・テープ)

このコマンドは，カセットテープにセーブされたデータと，メモリの内容とが一致しているかどうかを比較するコマンドです。Vコマンドは，次のように入力します。

v〔<ファイル名>〕⏎

動作は，Rコマンドとほぼ同じで，<ファイル名>を指定した場合は，指定したファイルとメモリの内容を比較し，<ファイル名>を指定しなかった場合は，最初に見つかったファイルと，メモリの内容を比較します。比較した結果，カセットテープの内容とメモリの内容が一致した場合は，コマンド入力待ち状態となり，一致しなかった場合は，“?”を表示してコマンド入力待ちになります。

(16) W(ライト・テープ)

メモリの内容をカセットテープへセーブするコマンドです。Wコマンドは次の形式で入力します。

w〔〈ファイル名〉〕，〈セーブ開始アドレス〉，〈セーブ終了アドレス〉[⏎]

例

8000番地から，8500番地の内容を“abc”というファイル名でカセットテープにセーブする場合

h〕wabc, 8000, 8500 [⏎]

(17) X(イグザミン・レジスタ)

Xコマンドは，CPUの全レジスタの内容の表示及び，変更を行うコマンドです。

x〔〈レジスタ名〉〕[⏎]

〈レジスタ名〉は，

a, b, d, h, a′, b′, d′, h′, s, p, x, y, i

です。a′のようにダッシュ記号がついているレジスタ名は，裏レジスタを指します。

〈レジスタ名〉を指定しない場合は，全レジスタの内容を表示して実行を終了します。〈レジスタ名〉を指定した場合は，そのレジスタの内容を表示し，新しい値の入力待ちになります。新しい値を入力して，リターンキー([⏎])を入力すると，コマンド入力待ち状態になり，スペースキーを入力すると，次のレジスタの処理へと移ります。新しい値を入力しない場合も，リターンキーを入力すれば，コマンド入力待ち状態となり，スペースキーを入力すると，次のレジスタの処理へと進みます。

各レジスタは，Ｉレジスタを除いて，すべてのレジスタは16bitのペアレジスタとして扱われます。ですから，8bit分のレジスタだけ変更する場合は，変更しない残りの8bit分のデータも入力してください。また，フラグレジスタで，フラグを立てないときは“－”を入れます。

(18) HELPキー([**HELP**])

CTRL-A

[**HELP**]キーまたはCTRL-Aを入力すると，モニタで使用できるコマンドの種類とそのコマンドのパラメータがディスプレイに表示されます。

(19) CTRL-B(リターン)

CTRL-Bを入力すると，N_{88}-BASIC(N_{88}DISK-BASIC)モードに戻ります。

(20) CTRL-D(ダンプ・ディスク)(N_{88}DISK-BASIC)

CTRL-Dは，フロッピィディスクの内容をディスプレイに表示させるコマンドです。

CTRL-D 〈ドライブ#〉〔,〈サーフェス#〉〕,〈トラック#1〉,
〈セクタ#1〉〔,〈トラック#2〉,〈セクタ#2〉〕

ドライブ#，〔サーフェス#〕，トラック#1，セクタ#1
……表示開始セクタの指定

トラック#2，セクタ#2　　　……表示終了セクタの指定

〈トラック#2〉，〈セクタ#2〉を指定しない場合は，

〈表示開始セクタ〉＝〈表示終了セクタ〉

となり，表示開始セクタだけ表示します。

CTRL-Dコマンドを実行中，実行を中止したい場合は，**STOP** キーまたは，CTRL-Cを入力します。画面への表示を一時中断したい場合は，CTRL-Sを入力し，再開するときは，**STOP** キーおよびCTRL-C以外のキーを入力します。

〈サーフェス#〉は，両面倍密度のフロッピィディスクユニットを使用しているときに指定します。ただし，モニタでは，両面にまたがってアクセスすることはできません。(CTRL-R，CTRL-Wも同じです。)

(21) CTRL-R(リード・ディスク)(N_{88}DISK-BASIC)

CTRL-Rコマンドは，フロッピィディスクから，メモリへデータをロードするコマンドです。

CTRL-R〈ドライブ#〉,〔,〈サーフェス#〉〕,〈トラック#〉,
〈セクタ#〉,〈開始アドレス〉,〈終了アドレス〉

このコマンドは，〈開始アドレス〉と〈終了アドレス〉で指定されたメモリ領域に格納するデータを，〈ドライブ#〉〔,〈サーフェス#〉〕,〈トラック#〉，〈セクタ#〉で指定されたセクタからロードします。ですから，大きなメモリ領域にデータをロードする場合は，複数のセクタ，トラックからデータをロードします。

例

h〕∧ r0, 9, 3, 1000, 100f

h〕

ドライブ0の第9トラックの第3セクタから，16バイトのデータをロードしてきて，1000番地から100f番地に格納します。このとき，メモリ領域の1010～10FF番地の内容は変更されません。

(22) CTRL-W(ライト・ディスク)

CTRL-Rコマンドと逆の操作を行うコマンドで，メモリの内容をフロッピィディスクへセーブします。

CTRL-W 〈ドライブ#〉，〔，〈サーフェス#〉〕，〈トラック#〉，〈セクタ#〉，〈開始アドレス〉，〈終了アドレス〉

〈開始アドレス〉と〈終了アドレス〉は，セーブするメモリ領域を指定します。この時，指定する領域は，1セクタ単位(256バイト)である必要はありません。ただし，フロッピィディスクは，セクタ単位で指定しますから，最低1セクタを使用します。

注　意

標準フロッピィディスクでは，CTRL-D, CTRL-R, CTRL-Wコマンドを，0トラックに対して行うことはできません。

13. 7　N-BASICのモニタ

N-BASICまたはDISK-BASICモードで，

mon ⏎

と入力すると，N-BASICのモニタに入ります。このとき，コマンド入力待ちを表す記号には“＊”が使われています。

N-BASICのモニタには次のコマンドがあります。

コマンド	内　　容	コマンド	内　　容
S	セットメモリ	**W**	ライトテープ
D	ディスプレイメモリ	**G**	ジャンプ
L	ロードテープ	**CTRL-B**	リターンベーシック
LV	ロードベリファイテープ	**TM**	テストメモリ

(1)　S(セットメモリ)

メモリの内容を変更します。

＊SXXXX ⏎ (XXXX＝アドレス)

〈入力可能なKEY〉

・変更データ　2桁の16進数(00～FF)

・アドレスを－1 CTRL-Hまたは INS DEL

・処理の終了 ⏎または STOP

・次のアドレスのデータ表示　スペースキー

(2)　D(ディスプレイメモリ)

メモリの内容を表示します。

＊DXXXX〔，YYYY〕 ⏎

(XXXX＝スタートアドレス，YYYY＝エンドアドレス)

表示を途中で中断したい場合は ESC を入力します。再開するときは，もう一度 ESC を入力します。中止する場合は，STOP を入力します。

(3)　W(ライトテープ)

メモリの内容をカセットテープへ書き込みます。

***W XXXX, YYYY** ⏎

（XXXX＝スタートアドレス，YYYY＝エンドアドレス）

(4) L（ロードテープ）

テープに書かれているデータをメモリに読み込みます。

***L** ⏎

(5) LV（ロードベリファイテープ）

テープに書かれている内容とメモリの内容を比較します。

***LV** ⏎

(6) G（ジャンプ）

あるアドレスにジャンプするときに使用します。

***G XXXX** ⏎ （XXXX＝ジャンプアドレス）

(7) CTRL-B（リターンベーシック）

モニタからN-BASIC（DISK-BASIC）に戻すときに使用します。

***CTRL-B**

(8) TM（テストメモリ）

メモリをテストするときに使用します。ただし，N-BASICで使用している部分だけしかテストしません。テストが無事に終了すれば，N-BASIC（またはDISK-BASIC）がスタートします。もし，エラーがあれば，ブザーが鳴りっぱなしになって処理は停止します。

参 照

各コマンドについての詳細は，「PC-8001ユーザーズマニュアル」を参照してください。

13.8　機械語のプログラムを使うときの注意

BASICだけを使ってプログラミングする場合(BASICのPOKE, OUT, USERなど機械語を扱う命令を使わないとき)は，ユーザは，ユーザーズエリア，ワークエリアなどのメモリの管理をする必要は全くありません。ユーザは，Out of Memoryエラーが出たときに，メモリが足りなくなったことを知るぐらいで，通常は，メモリがどのように使われているかを気にする必要はないのです。一方，機械語のプログラミングをする場合，または，市販のソフトウェアで機械語が含まれているのを使用する場合は，注意が必要です。

機械語のプログラミングの経験のある方はご存知のように，不完全な機械語のプログラムを実行させた場合，PC-8801MKⅡの制御がきかなくなってしまう(キー入力ができなくなる)ことがよくあります。機械語のプログラムは，BASICの管理下にありませんから，エラーがあったとしても，エラーメッセージを出力してコマンド入力待ちの状態に戻ってくることはないからです。(ユーザから見れば，プログラムの誤りでも，μPD780は，誤りと思わないで,正直に実行を続けます。) PC-8801MKⅡの制御がきかなくなってしまうと，機械語のプログラムがどのように実行されているかは見当もつきません。突然コールドスタートがかかってしまうこともあれば，メモリの内容をめちゃくちゃに書き替えて，BASICのワークエリアや，BASICのプログラムが書かれている部分，あるいは，機械語プログラム自身をこわしてしまうことも考えられます。

このような場合は，ウォームスタート(できなければコールドスタート)をかけます。ウォームスタートができても，上記のように，メモリが書きかえられた可能性がありますから，メモリがもとの状態に戻る，という保障はありません。

したがって，機械語のプログラムを作成する場合は，メモリ上のBASICのプログラムなどが，失われてもいいように，必ず，セーブしておいてから実行させるようにしてください。また，機械語のプログラムを使ったあと,他の作業(別のBASICのプログラムを使うときなど)を始める前に，コールドスタートを行って，メモリのイニシャライズをすることをお勧めします。

第14章

第14章
ターミナルモード

この章では，PC-8801MKⅡをターミナルとして使う方法を説明します。PC-8801MKⅡをターミナルとして使うことによって，遠く離れたコンピュータとのやりとり，別のパーソナルコンピュータとのネットワーク，電話回線を通して各種のコンピュータサービスを受けたり，など幅広い応用ができます。

（解　説）

PC-8801MKⅡは，ターミナル用のインタフェースとして，RS-232C規格のインタフェースを持っています。RS-232C規格はEIA(Electronic Industries Association)で規定された，シリアルデータのインタフェース規格で，日本でもJIS C6361-71として制定されています。RS-232C規格はターミナル用のインタフェースの他に，いろいろなI/O機器のインタフェースとして採用されています。

（注　意）

PC-8801MKⅡと他のコンピュータとを接続する場合は，接続するコンピュータの仕様を十分に調べた上で御使用ください。

14. 1　ターミナルモードへの切り換え

PC-8801MKⅡを，他のコンピュータ(以後，ホスト・コンピュータと呼びます。)の端末として，送られてきたデータを表示する場合は，ターミナル・モードでの使用が適しています。

PC－8895
PC－8801MKⅡ
コンピュータ
音響カプラ

PC-8801MKⅡを，ターミナルモードにするには，次の3つの方法があります。

1. N_{88}-BASICモード(または，N_{88}DISK-BASICモード)からterm文を入力します。
2. N_{88}-BASICモード(または，N_{88}DISK-BASICモード)からnew on文を入力します。(詳しくは，BASIC REFERENCE MANUALを参照してください。)
3. PC-8801MKⅡ本体前部のディップスイッチをターミナルモードに設定してから，電源を「ON」にします。

PC-8801MKⅡをターミナルモードでしか使用しない場合は，3の方法が便利でしょう。

14．1．1　ボーレートの設定

PC-8801MKⅡ本体前部のジャンパスイッチにより，ボーレート(伝送速度)を切り替えることができます。出荷時は9600ボーにセットされていますから，ホストコンピュータの仕様に合わせて設定してください。

ジャンパースイッチ	ボーレート
1	75
2	150
3	300
4	600
5	1200
6	2400
7	4800
8	9600

RS-232C

1　2　3　4　5　6　7　8

(注　意)

スイッチ1～8は，この中のどれか1つを選択します。2個以上を同時に選択しないでください。

14.1.2　term文を使用してターミナルモードに切り換える方法

N88-BASICまたは，N88DISK-BASICをスタートさせ，次のterm文を入力します。また，ターミナルモードから抜けるときは[SHIFT]＋[STOP]を入力します。各パラメータは大文字を使用して下さい。

term "com:〈パリティ〉〈ビット長〉〈ストップ・ビット〉〈Xパラメータ〉〈Sパラメータ〉"〔,〔〈通信モード〉〕〔,〈スタック長〉〕〕

〈パリティ〉：パリティチェックの指示を行います。

偶数パリティ(even parity)　…………E

奇数パリティ(odd parity)　…………O

パリティチェックなし(non parity)…………N

〈ビット長〉：データのビット長を表します。

7ビット…… 7

8ビット…… 8

〈ストップ・ビット〉：ストップ・ビットの長さを表します。

1ビット …… 1

1.5ビット…… 2

2ビット …… 3

〈Xパラメータ〉：Xパラメータの有効，無効の切換えを行います。

X　……Xパラメータ有効

省略……ディップ・スイッチの値(「14.1.3」参照)

無効……N

「Xパラメータ」を有効にすると，PC-8801MKⅡのデータ転送用バッファの4分の3までデータが入ると，自動的にホストコンピュータに"CTRL-S"コードを出力し，データ転送の一時停止を要求し，4分の1になると"CTRL-Q"を出力して転送の再開を要求します。

〈Sパラメータ〉：Sパラメータの有効，無効の切り換えを行います。

有効……S

無効……N

省略……ディップ・スイッチの値(「14.1.3」参照)

「Sパラメータ」を有効にすると，7 bitモードでカナを送受信することができます。つまり，SOコードがくると，それ以後のデータをカナとみなし，SIコードがくると，それ以後のデータを英数字とみなします。

〈通信モード〉：全二重，半二重の指定

全二重……F

半二重……H

省　略……ディップスイッチの値(「14.1.3」参照)

注　意

「全二重モード」では，データの送信，受信が同時に行えます。「半二重モード」では，送信，受信のどちらかが行われている時には，1行分のデータの送信(受信)が終るまで(“CR”コードが来るまで)，次の受信(送信)は行えません。

〈スタック長〉：ターミナルモードでも，ホストコンピュータから送られてきたBASICのコマンドを実行することができますが，その時，使用する演算スタックの大きさを指定します。省略すると1024バイトが設定されます。

例

term “com:E72X”, H

偶数パリティ，データ長7ビット，ストップビット1.5ビット，Xパラメータ有効，半二重モードのターミナルモードになります。

term文のパラメータは次のように省略することができます。

term “com:”

term “△”

この場合は，各パラメータともディップ・スイッチの値もしくは，標準値(スタック長)がセットされます。

ターミナルモードになっているPC-8801MKⅡからホストコンピュータにBreak信号を送る場合は，STOP キーを入力します。

14. 1. 3 ディップスイッチでターミナルモードへ切り換える方法

(1) まず，PC-8801MKⅡおよび，周辺機器の電源を「OFF」にします。

(2) PC-8801MKⅡ前面のディップスイッチSW 1の1を「OFF」(下向き)にし，2を「ON」(上向き)にします。

SW1　1 2 3 4 5 6 7 8　　SW2　1 2 3 4 5 6 7 8

(3) 他のディップスイッチで，各種モードを設定します。

ディップスイッチ		ON(上向き)	OFF(下向き)
SW1	5	Sパラメータが有効	Sパラメータが無効
	6	DELコードを処理する。	DELコードを無視する。
SW2	1	パリティチェック有り	パリティチェック無し
	2	偶数パリティ	奇数パリティ
	3	データ長が8ビット	データ長が7ビット
	4	ストップビットが2ビット	ストップビットが1ビット
	5	Xパラメータが有効	Xパラメータが無効
	6	半二重モード	全二重モード

(4) 各装置の電源を「ON」にします。これで最初からターミナルモードがスタートします。

例

偶数パリティ，8ビット長，ストップビット長が2ビット，Xパラメータ有効，Sパラメータ有効，半二重モードに設定する場合。

(term “com : E83XS”, Hと同じです。)

SW1　　　　　　　　SW2

1 2 3 4 5 6 7 8　　1 2 3 4 5 6 7 8

ディップスイッチでターミナルモードに入った場合も，

SHIFT+STOP

で，ターミナルモードから抜け出すことができます。

14.2　ターミナルモードの機能

14.2.1　ファンクションキー

N88-BASIC(N88DISK-BASIC)のターミナルモードになったとき，ファンクションキーのf・6～f・10は次のように定義されています。

f・6 literal

このキーを入力すると，コントロールコードが実行されず，文字として表示されます。ただし，リターンとラインフィードは，実行されます。

f・7 half/full

全二重，半二重の切り換えを行います。初期状態はパラメータまたはディップ・スイッチでセットされています。

f・8 LPT enable

画面への出力をプリンタへも出力します。もう一度入力すると出力を停止します。初期状態は出力しないモードにセットされています。

f・9 copy buffer

画面以外のバッファの内容をプリンタへ出力します。

f・10 LPT feed

プリンタに1行フィードするコードを出力します。

f・1～f・5は，ターミナルモードになる前に，BASICモードで定義されたものになります。

14.2.2 スペシャル・エスケープ・シーケンス

ターミナルモードに切り換えた場合は，BASICのモードから抜け出てしまいますが，〈スペシャル・エスケープ・シーケンス〉を利用すれば，BASICの命令を実行できます。

〈スペシャル・エスケープ・シーケンス〉は，ホストコンピュータから後に示すようなエスケープコードを送ることにより，ターミナルモードのPC-8801MKⅡに接続されているディスプレイの画面の表示桁数を変えたり，文字の色を変えることができます。また，BASICの命令を実行するだけでなく，プリンタへのエコーバックの切換えなども，この機能を用いれば簡単に行えます。次に個別の機能を説明します。

(1) **ESC@** ： プリンタへの出力(エコーバック)を開始します。

(2) **ESC A** ： プリンタへの出力(エコーバック)を解除します。

(3) **ESC"** ： PC-8801MKⅡのキー入力を有効にします。

(4) **ESC#** ： PC-8801MKⅡのキー入力を無効にします。

(5) **ESC!** ： ホストコンピュータへIDコードを出力します。

(6) **ESC*** ： 現在スクロールされている範囲をクリアします。

(7) **ESC Y** ： 現在のカーソル位置からスクロールされている最下位行までをクリアします。

(8) **ESC△** ： ターミナルのイニシャライズを行います。(termコマンドを実行した直後と同じ状態にします。)

(9) **ESC= 〈X-pos〉〈Y-pos〉**
: カーソル位置を指定します。（$20_{(16)}$のバイアスがかかります。）

例

カーソルの位置を（3，8）に移動させるには，

3 →$23_{(16)}$='#'

8 →$28_{(16)}$='C'

ですから，

ESC=#C

とします。

(10) **ESC T**
: 現在カーソルがある行を1行消去します。

(11) **ESC> BASIC STATEMENT**
: BASICステートメントを実行します。BASICの実行結果は，ターミナルとなっているPC-8801MKII側に表示されます。

(12) **ESC . BASIC STATEMENT**
: BASICステートメントを実行します。ただし，実行結果は表示されません。

(13) **ESC <BASIC STATEMENT**
: BASICステートメントを実行します。実行結果の表示はホストコンピュータ側だけです。

14.2.3　リモート・BASIC・プロトコル

「14.2.2　スペシャル・エスケープ・シーケンス」で説明した機能の中で，

(11)　**ESC>BASIC STATEMENT**

(12)　**ESC . BASIC STATEMENT**

(13)　**ESC<BASIC STATEMENT**

の3つを「リモート・BASIC・プロトコル(remote BASIC protocol)」と言います。これは，ホストコンピュータから，ターミナルとなっているPC-8801MKⅡのBASICの機能が利用できるようになっています。ただし，この機能にはいくつかの制約があります。

(i)　使用できないBASICステートメント

　リモートBASIC・プロトコルでは，メモリ内にプログラムを作成したり，メモリ内のプログラムの変更，削除は行えません。また，メモリ内の変数エリアを直接扱う，ステートメント(例えば，fre(0), clear, eraseなど)も実行できません。さらに，同じバッファを使って入出力を行うファイルをOPENさせることはできません。(com:, cas:)

(ii)　実行結果の表示

　print文やprint using文など実行結果を画面に表示させる命令では，表示をターミナル側に行うか，ホストコンピュータ側で行うかを選択できます。すなわちエスケープコード(ESC)の後の符号(>. <)によって指定できるわけです。しかし，グラフィック命令は，この符号の指定にかかわらず，必ずターミナルとなっているPC-8801MKⅡに接続されているディスプレイに実行結果を表示します。

(iii)　使用変数の制限

　リモート・BASIC・プロトコルでは，使用できる変数エリアが定っています。また，一度使用するとそのエリアは二度と別の変数には使えませんから，多くの変数に値を代入していくと，途中で代入できなくなってしまいます。リモート・BASIC・プロトコルで使用する変数は，必要最小限にとどめてください。

例

リモート・BASIC・プロトコルの利用例

- ターミナルにグラフを表示させる。

```
ESC>screen 0,0
ESC>circle(100,50),100,7
ESC>a =100: b =100: c =200: d =50
ESC>line(100,50)-(a, b), 7
ESC>line(100,50)-(c, d), 7
ESC>paint(120,70), 1, 7
```

のようにホストコンピュータからコードを送ると，ターミナルに鮮やかなグラフが簡単に表示できます。

- N88DISK－BASICからターミナルモードに入った場合

```
ESC>open "2:TEST 1" for output as #1
ESC>print#1,100
```

のように，ターミナルになっているPC-8801MKⅡに接続されているフロッピィディスクにデータをセーブできます。

14.3 N-BASICのターミナルモード

(1) N-BASICにおけるターミナルモードの基本動作

(i) RS-232Cポートから入力した文字をディスプレイに表示します。

(ii) キーボードから入力された文字をRS-232Cポートに出力します。

N-BASICから，ターミナルモードに入るには，次の命令を実行します。

```
term | a | , | 0 | , | 0 | , | 0 |
     |   |   | 1 |   |   |   |   |
     | j |   | 2 |   | 1 |   | 1 |
```

- 第4パラメータ：{ 1＝オートL/ Fをする / 0＝オートL/ Fをしない }
- 第3パラメータ：{ 1＝ボーレートファクタ16 / 0＝ボーレートファクタ64 }
- 第2パラメータ：{ 0 ＝パリティビットをつけない / 1＝奇数パリティ / 2＝偶数パリティ }
- 第1パラメータ：{ a ＝8bitモード / j＝7bitモード }

ボーレートファクタとはμPD8251のボーレートファクタそのものです。オートL/Fを指定すると，CRコードを受信したときに，自動的に改行も行います。

(2) ターミナルモードでのファンクションキー

ターミナルモードでは，ファンクションキー6～10は，以下のように特別な意味を持っています。

f・6：コントロールコードの表示をするか否か

f・7：半二重/全二重の切り換え

f・8：LPに出力するかどうか

f・9：スクリーンのLPへのコピー(ページプリント)

f・10：LPのラインフィードを行う。

これらのうち，f・6～f・8は押すたびに現在と逆の状態にかわり

ます。(すなわち，２回押すともとに戻ります。)

ターミナルモードにはいった直後は，ｆ・６～ｆ・８は，それぞれ次の状態になっています。

ｆ・６：コントロールコードを表示しない

ｆ・７：全二重(オートエコーしない)

ｆ・８：LPに出力しない。

ターミナルモードから，N-BASICにもどるには，GRPH-B(**GRPH**キーを押しながらＢを押す)を使います。ターミナルモードに入ってもBASICプログラムはこわれません。

第15章

第15章
拡張命令

15. 1　拡張命令パッケージ

15. 1. 1　概　　要

拡張命令パッケージは，N_{88}DISK-BASIC(N_{88}-BASIC)にタートルグラフィック機能，サウンド機能などを追加するユーティリティプログラムです。

このユーティリティプログラムを使うことにより，以下の命令が使えるようになります。

(1)　CMD TURTLE
　LOGOライクなタートルグラフィック機能を実現するステートメント

(2)　CMD SING
　本体内蔵スピーカを利用して音楽を演奏するステートメント

(3)　CMD CLS
　高速画面クリアステートメント

(4)　CMD TEXT ON/OFF
　テキスト画面のON/OFFを制御するステートメント

(5)　CMD CUT
　拡張命令パッケージを切り離し，専有していたエリアをフリーエリアとして開放する

拡張命令パッケージは，機種によって，次の2つのものが用意されています。

(1)　ディスク版拡張命令パッケージ(model 20, model 30)
　本体添付の「N_{88}DISK-BASICシステムディスク」の中に入っています。

拡張命令パッケージは，以下の２つのファイルです。

"@**load. n88**"‥BASICファイル

"@**exst＊** "…機械語ファイル

(2)　カセット版拡張命令パッケージ(model 10)

本体添付の「カセットテープ」の中に入っています。(Ｂ面)カセット版は，以下の１つのファイルです。

"@**exst**"…BASICファイル

15.1.2　拡張命令の追加

N88DISK-BASIC(N88-BASIC)に拡張命令を追加するには，次の手順で行います。

・ディスク版拡張命令パッケージ

(1)　N88DISK-BASICを起動します。

(2)　拡張命令パッケージの入っているフロッピィディスクをドライブ番号１のフロッピィディスクドライブにセットします。

(3)　"@load. n88"をロードします。

load "@load. n88" ⏎

(4)　プログラムを実行します。

run ⏎

これにより，機械語ファイルがロードされ，拡張命令の初期化が行われます。

(5)　次のように表示されれば終了です。

```
The following statements are available.
  1.  CMD TURTLE
  2.  CMD SING
  3.  CMD CLS
  4.  CMD TEXT ON/OFF
  5.  CMD CUT
Please enjoy new features.
Ok
```

・**カセット版拡張命令パッケージ**

(1) N_{88}-BASICを起動します。

(2) 本体添付(model 10)のカセットテープから，プログラムをロードします。

load "cas2 ：@exst" ⏎

(3) プログラムを実行します。

run ⏎

(4) 次のように表示されれば終了です。

```
The following statements are available.
  1. CMD TURTLE
  2. CMD SING
  3. CMD CLS
  4. CMD TEXT ON/OFF
  5. CMD CUT
Please enjoy new features.
Ok
```

(5) これで，N_{88}-BASICが拡張され，cmd命令が使えるようになります。

15. 1 . 3 拡張命令の使用上の注意

(1) 拡張命令の使用エリア

拡張命令の処理ルーチンは，RAM上にロードされます。

拡張命令を追加した場合のメモリマップは，次のとおりです。

アドレス	エリア
FFFF	ワークエリア
E600	拡張命令処理ルーチン
DC00	フリーエリア
XXXX (不定)	ファイルバッファ (N_{88}DISK-BASIC)
8400	

したがって，この場合は，DC00H番地から，E5FFH番地を使うこ

とはできなくなります。特に，機械語を使ったプログラムを実行するときは注意が必要です。

(2) ディップスイッチの設定

拡張命令の中で，CMD SINGを使う場合は，ディップスイッチを次のように設定してください。

SW1

1	2	3	4	5	6	7	8	
								ON
							■	OFF

もし，ディップスイッチ1-8がONになっていると，CMD SINGを実行しても音が出なくなります。

(3) ウォームスタート

拡張命令パッケージが追加された状態でウォームスタートを行わないでください。ウォームスタートを行った後の拡張命令の動作は保障されません。

15.2 拡張命令の各機能

15.2.1 タートルグラフィック機能

今までは図形を描く時に，LINE命令やCIRCLE命令を使ってきました。このPC-8801MKⅡでは新しくタートルグラフィック機能が追加されより簡単に速く描けるようになっています。

画面に現われる小さな三角形をタートル(亀という意味)と呼びます。そのタートルを前後に進めたり，向きを変えることによって図形を描いていきます。タートルを動かす場合はサブコマンドストリングを用います。サブコマンドの詳しい説明は「BASIC REFERENCE MANUAL」を参照して下さい。

タートルを実際に動かす前に，タートルを画面の中央に表示させるため，次のようにタイプしそのあとにリターンキーを押して下さい。

POINT(320, 100)

“OK”が表示されたら，タートルを画面中央に設定する準備が出来たことになります。上記の動作をせずにタートルを表示させた場合，画面中央に

タートルが表示されないことがあります。POINT命令については「BASIC REFERENCE MANUAL」を参照して下さい。

タートルを動かすにはCMD TURTLE命令を使います。CMD△TURTLEをタイプしたあとにダブルクォーテーション(")をタイプします。(SHIFTキーを押しながら2"フを押して下さい。)そのあとにサブコマンドでタートルの動きを指定してダブルクォーテーションで閉じます。リターンキーを押すとタートルはあなたの指定した通りに動きます。次にいくつかのサブコマンドを使ってタートルを動かした例をあげましょう。

CMD TURTLE "ST"

タートルを画面に表示します。

CMD TURTLE "FD50"

50ステップ前に進めます。

CMD TURTLE "RT90"

右へ90度回転させます。

CMD TURTLE "FD100LT45"

100ステップ前進し左へ45度回転させます。

CMD TURTLE "BK50HD0"

50ステップ後退しタートルを０度の方向に向けます。

CMD TURTLE "PUFD30"

タートルのペンを上げ30ステップ前進させます。軌跡は残しません。

CMD TURTLE "PDPC5FD30"

タートルのペンを下げ色を水色に指定し30ステップ前進させます。

CMD TURTLE "HT"

タートルを消します。

上記の他にも次に示したサブコマンドを使用してタートルを動かすことができます。

CMD TURTLE "MV320, 100"

タートルをスクリーン座標のX方向320，Y方向100に移動させます。

CMD TURTLE "SX400"

タートルをスクリーン座標のX方向400に移動させます。

CMD TURTLE "SY150"

タートルをスクリーン座標のY方向150に移動させます。

CMD TURTLE "CP1"

640×200モードでX方向の補正をします。CP 1 の場合CMD TURTLE "RP 4 [FD50RT90]"を実行させると正方形を作りますが CR0の場合正方形にはなりません。 初期状態はCP1となっています。

タートルを動かす前にPOINT命令でタートルを画面中央に表示させる準備を行いましたがCMD TURTLE命令を使っても同様に出来ます。

CMD TURTLE "PUMV320, 100PD"

次にRPサブコマンドを使って図形を描いた例をあげます。その前に画面をクリアして下さい。方法はCMD△CLS△ 2 とタイプしてリターンキーを押します。ここで注意していただきたいのですが他のグラフィック命令(たとえばCMD CLS 2等)を実行する時は必ずタートルを消してから行うようにして下さい。タートルを表示させたままで他のグラフィック命令を実行し再びCMD TURTLE文を実行するとゴミがでることがあります。注意して下さい。

RPサブコマンドは [……] の中のサブコマンドを指定した回数を繰り返してタートルに図形を描かせます。次に簡単な例をあげます。

CMD TURTLE "RP 4 [FD50RT90]"

タートルを50ステップ前進させ右に90度回転させる動作を4回繰り返すと上の図のような正方形ができます。

上の例を応用してもう少し複雑な図形を描いてみましょう。

CMD TURTLE "RP18[RP 4[FD50RT90]RT20]"

タートルを50ステップ前進させ右に90度回転させる動作を4回繰り返し正方形を描いたあと,右に20度ずつ回転させながら,正方形を描く動作を18回繰り返します。

RPサブコマンドにおいてネスティング([ ]の中にまた[ ]を入れること)は8レベルまでできます。

CMD TURTLE命令は直線だけでなく円や曲線も描くことができます。円の場合を例にとってみましょう。

CMD TURTLE "RP60 [RT6FD5]"

タートルを右に６度回転させ５ステップ前進させていく動作を60回繰り返すと円が出来上がります。６度×60回＝360度。ゆえにタートルは１回転したことになります。

上記の例で半円を作るなら繰り返しの回数を30回にするとタートルは180度回転し半円となりますし，また¼円を作るならさらに回数を減らして15回にすればタートルは90度回転して¼円を作るわけです。

繰り返しの回数を変化させず，RTのパラメータを変化させることによっても半円が出来ます。

円の大きさを変えたければFDのパラメータを変えればよいわけです。

パラメータは変数名を使用した場合のみ倍精度実数型まで扱えますが，小数点第１位で四捨五入されます。変数名を使用しない時は整数型を用いて下さい。扱える数値の範囲は－32768～＋32767です。

以上，タートルグラフィック機能の扱い方について簡単に説明してきました。タートルグラフィック機能だけを使って図形を描いても楽しめますが他のグラフィック命令と一緒に使ったり，またsin，cos等の関数と組み合わせたりすれば，一層楽しめることでしょう。

Note **STコマンドを使ってタートルを表示させた状態で，他のグラフィック命令（CLS 2）などを使うと，次にCMD TURTLEを実行した際，ゴミが出ることがあります。CMD TURTLEと他のグラフィック命令を混ぜて使う場合は，HTコマンドでタートルを消して使うようにしてください。**

15.2.2　サウンド機能

PC-8801MKⅡには新しい機能が２つあります。１つは前項に書かれているタートルグラフィック機能でありもう１つはこれから説明するサウンド機能です。

PC-8801ではユーザーがゲームプログラムを作成し装飾音を挿入しようと考えてもBEEP音しか使用できませんでした。PC-8801MKⅡではゲームプログラムにサウンド機能を追加することによって装飾音を取り入れゲームをより効果的に演出することができます。

また音楽は好きだが楽器は苦手という方でしたらキーボード入力をするだけで簡単に音楽を楽しめます。

音はCMD SING文のサブコマンドストリングによって指定します。このCMD SING文は一音演奏が可能です。

音量調節は本体前面右下の扉を開けると左側に「VOLUME」と書かれたつまみがあります。右にまわすと音が大きくなり左にまわすと音が小さくなります。調節する時はマイナスドライバーを使用して下さい。

ここで簡単にオクターブ，音符，音符と休符の長さの設定について説明します。

オクターブを設定するサブコマンドは「On」を用います。オクターブの対応は次の通りです。

n＝3　n＝4　n＝5　n＝6

音域は全部で４オクターブありますが1番高いオクターブのＢ♯(シ♯)は音が出ません。

音符は「C, D, E, F, G, A, B」で指定をします。アルファベットではしにくいという方は次の図を参照して下さい。

C+　D+　F+　G+　A+

ド　レ　ミ　ファ　ソ　ラ　シ

C　D　E　F　G　A　B

半音上げる時(♯)は＋，半音下げる時(♭)は－をつけます。音符の長さはそのあとで指定します。

「Ln」で音符の長さ，「Rn」で休符の長さを指定します。長さの対応は次の通りになります。

n＝1	n＝2	n＝4	n＝8	n＝16	n＝32
全音符	2分音符	4分音符	8分音符	16分音符	32分音符
全休符	2分休符	4分休符	8分休符	16分休符	32分休符

尚サブコマンドの詳しい説明は「BASIC REFERENCE MANUAL」にも掲載されています。そちらも参照して下さい。

では実際に音を出してみましょう。CMD△SINGとタイプしたあとにダブルクォーテーション(")をタイプします。(SHIFT キーを押しながら 2" を押して下さい。)サブコマンドによって音を指定したあと，ダブルクォーテーションで閉じます。そのあとにリターンキーを押すとあなたの指定した通りに演奏してくれます。

簡単なところから始めましょう。

CMD SING "X0T100O4L4CDE"

新しく，出てきたサブコマンドも含めて説明していきましょう。「X0」でテキスト画面の表示を制御します。「X0」の場合CMD SING文を実行している間テキスト画面は表示されません。ただしグラフィック画面は「X0」の場合でも表示可能です。「X1」の場合はCMD SING文を実行している間でもテキスト画面の表示が可能ですが澄んだ音が出ませんので注意して下さい。

「T」はテンポです。上記の場合は100と設定してあります。１分間に演奏する４分音符の数を100としてテンポを設定します。

「O」でオクターブを４に設定します。音符の長さはＬ４で指定されている通りに４分音符となります。"CDE"はそれぞれ４分音符で演奏されます。音符のあとにも長さを指定することができます。たとえば上記の条件で"CD８E"の場合,"C" と "E" は4分音符で演奏されますが "D" は８分音符で演奏されます。長い曲を演奏する時は，よく出てくる音符の長さをＬサブコマンドを使って最初に指定しておき，違う長さの時は音符のあとにパラメータを使って長さを設定する方法が便利でしょう。

上記の例を使ってテンポを変えたり，オクターブや音符の長さを変えるなどして，どのように違ってくるか調べてみてもよいでしょう。

テンポは48から255まで，オクターブは３から６まで，音の長さは１から32まで演奏が可能です。次は少し長めに打ち込んでみましょう。

CMD SING " X０T120O５L８GF＋GD４．O４B４．AG＋AO５E４．E４．DF＋EEDCCO４BAB４．B４．B４．"

ここで気をつけていただきたいことは最初に指定したオクターブより低い音あるいは高い音が出てきた場合は，そこでまたオクターブを指定しなければならないということです。上記の例で見ていくと"B4."の前に"O4"とオクターブが指定されています。もしオクターブをそこで指定しなかったらどうでしょう。"O5"の時のBの音が実行されてしまいます。またここで指定された"O4"は新たにオクターブを指定しない限りそのあとに続く音符全てに実行されます。"B4.AG＋A"は"O4"で演奏されますが次に"O5"と指定されていますのでそのあとに続く音符は"Ｏ５"で実行されることになります。

"F＋" はF♯のことです。"4." は符点4分音符（♩.）のことです。

上記の曲を知っている方，あるいは楽譜を持っている方は練習のために続きを打ち込んでみて下さい。その時は行番号をつけてプログラムモードの形で打ち込んでいくとよいでしょう。

ここで先程ダイレクトモード(行番号をつけないでキーボードから入力した命令をすぐに実行させること。今までの例はダイレクトモードによって実行させていました。)で打ち込んだものを行番号をつけてプログラム

モードにしてみましょう。

10△CMD△SING "X0T120O5L8"

20△CMD△SING "GF+GD4.O4B4.AG+AO5E4.E4."

30△CMD△SING "DF+EEDCCO4BAB4.B4.B4."

おのおのの行が入力し終わったら必ずリターンキーを押して下さい。全ての行を入力し終わったら"run"とタイプしてリターンキー を押して下さい。ダイレクトモードで打ち込んだ時と同様に演奏されます。

CMD SING文もCMD TURTLE文と同様にRPサブコマンドが使用できます。

CMD SING "X0T200O5L8RP3 [CEGE] C2"

RPコマンドは [ ] の中の音符を指定した回数繰り返して演奏します。この場合は"CEGE"を3回繰り返し演奏します。

CMD SING文ではパラメータに変数名を使用することもできます。倍精度実数型まで扱えますが小数点第1位で四捨五入されます。変数名を用いない時は整数型で指定して下さい。

以上，サウンド機能について説明してきましたが，長い曲等を打ち込む場合はREAD文，DATA文等を使用してプログラム文を作成するときれいなプログラム文ができます。

15.2.3 その他の機能

PC-8801MKⅡではタートルグラフィック機能，サウンド機能の他にもう2つ機能を持っています。それぞれについて説明していきましょう。

(1) CMD CLS

現在アクティブな画面，または指定したグラフィック画面をクリアします。CLS文より実行速度が速くなります。

書式は

CMD CLS [〈機能〉]

です。機能には1，2，3および9～15までの値を入れます。

機能を省略した場合はCMD CLS 1とみなされます。

次にそれぞれの働きを説明します。

1) テキスト画面をクリアします。

2) 現在アクティブなグラフィック画面をクリアします。カラーモードにおいては現在表示されている画面がアクティブとなります。白黒モードにおいてはscreen文の第3パラメータで指定された画面がアクティブとなります。CLS文ではグラフィック画面のビューポート内のみをクリアしますがCMD CLS 2ではビューポートは無視してクリアされます。LP(最後にグラフィック操作の行われた座標)は移動しません。

3) テキスト画面，グラフィック画面の両方をクリアします。

9～15) 指定したグラフィック画面のみをクリアします。画面がアクティブであっても，あるいはなくても働きは変わりません。

	PAGE 0(B)	PAGE 1(R)	PAGE 2(G)
9	○	×	×
10	×	○	×
11	○	○	×
12	×	×	○
13	○	×	○
14	×	○	○
15	○	○	○

○……クリアする
×……そのまま

CMD CLS 1あるいはCMD CLS 3が実行された時，テキスト画面ではconsole文によって指定されたスクロールウィンドウ内のみがクリアされます。

(2) CMD TEXT ON/OFF

テキスト画面の表示を制御します。

CMD TEXT ONの場合，テキスト画面を表示したまま，プログラムの実行を行います。

CMD TEXT OFFの場合，テキスト画面の表示を止めたまま，プログラムを実行します。それにより実行速度がアップされます。

ただしコマンド待ちに戻る前(プログラムの終了時，プログラム中でのエラー発生時)には，自動的にテキスト画面が表示されます。この機能は，CMD SINGのサブコマンドでもサポートされています。

注　意

テキスト画面がカラーモードでグラフィック画面が白黒モードのとき，CMD TEXT OFFを実行するとグラフィック画面も消えます。

15.2.4　拡張命令の切り離し

拡張命令パッケージを追加するとBASICのフリーエリアが減少します。大きなフリーエリアを必要とするプログラムを実行する場合は，拡張命令パッケージを切り離してください。また，機械語を使ったプログラムの中には，拡張命令パッケージと同じ番地を使っているものもありますので，このような場合も拡張命令パッケージを切り離して使ってください。(ディスクユーティリティプログラムも含まれます。)

拡張命令パッケージを切り離す命令としては，CMD CUT文が用意されています。

この命令は，拡張命令パッケージを切り離して，専有していた領域をフリーエリアとして開放するものです。

実行の方法は，ダイレクトモードで，

CMD CUT ⏎

と入力します。

この命令を実行すると，CLEAR，&HE5FFも同時に行なわれ，以後CMDステートメントは使用できなくなります。

付　録

付　　録

付録１ハードウェア概要

付録１．１　機能仕様

(1)　CPU

　メイン　CPU

　　μPD780C-1　　4MHz

　サブ　CPU

　　μPD780C-1(ディスク・コントロール)　4MHz

(2)　ROM

　メイン

　　N-BASICおよびモニタ　　32Kバイト

　　N_{88}-BASIC　　40Kバイト

　　スロット内増設可能　　最大56Kバイト
　　　　(８Kバイト×７バンク)

　サブ

　　ディスク・コントロール用　　２Kバイト

(3)　RAM

　メイン

　　ユーザーズメモリ　　64Kバイト

　　　N_{88}-BASIC動作時

　　　　テキストエリア　　32Kバイト

　　　　変数・ワークエリア・テキストVRAM　31Kバイト

　　グラフィック用VRAM　　48Kバイト

　　スロット内増設可能　　32K単位でバンク切り替え

　サブ

　　ディスク入出力バッファ・ワークエリア　16Kバイト

(4) 表示能力

テキスト表示

80文字×25行，80文字×20行

40文字×25行，40文字×20行

※上記のいずれかを選択可

リバース，ブリンク，シークレット(キャラクタ単位指定可)

カラー8色(キャラクタ単位に可能)

グラフィック表示

640×200ドット　3画面

640×400ドット　1画面(専用高解像度ディスプレイ使用時)

＊上記のいずれかの画面を選択

画面合成可(グラフィック，テキスト合成)

カラー8色(キャラクタ単位に可能)

カラーグラフィック表示

640×200ドット　1画面

カラー8色(ドット単位に指定可)

画面合成可(カラーグラフィック，テキスト合成)

※カラーグラフィック，グラフィック表示は，N_{88}-BASIC動作時

バックグランドカラー

8色指定可

ビデオ出力

R.G.Bセパレート出力方式(カラー)

コンポジットビデオ信号出力方式(輝度変調，モノクロ)

家庭用T.V.(T.V.アダプタ経由)に接続可

2画面独立表示可

(5) 漢字ROM

標準実装

文字構成	16×16ドット
文字種類	JIS第一水準の漢字(2965字)
	非漢字(約700種)
画面構成	40文字×20行(専用高解像度ディスプレイ使用時)

(6) キーボード

JIS標準配列準拠

テンキー，コントロールキー，5ファンクションキー，キャピタルロック可，HELPキー，COPYキー

セパレートタイプ(本体とカールケーブルにより接続)

(7) 拡張用スロット

3スロット(PC-8012，PC-8013，PC-8801，PC-8001mkII　上位コンパチブル)

(8) オーディオカセットインタフェース

600ボー/1200ボー

(9) 汎用I/O

入力2ビット，出力3ビット，割り込み1ch

(10) プリンタインタフェース

パラレルインタフェース(セントロニクス社仕様に準拠)

(11) シリアルインタフェース

RS-232C規格に準拠。割り込み/ポーリング制御可

75/150/300/600/1200/2400/4800/9600ボー

(12) 8インチフロッピィディスクインタフェース

本体内スロットに内蔵可

(13) ミニフロッピィディスク

本体内に実装可(最大2基)

拡張用ミニフロッピィディスク(PC-80S32用)インタフェース内蔵

(14) カレンダ時計

Ni-Cd電池でバックアップ

(15) 電源

AC100V±10%，50/60Hz　消費電力　46W(model 30)

(16) 使用条件

10～35℃，20～80%(但し結露しないこと)

(17) 外形寸法・重量

本体	350(W)×345(D)×125(H)mm	9.5kg (model 30)
キーボード	412(W)×195(D)×32(H)mm	1.4kg

付録1.2　ブロックダイヤグラム

付録1.3 メモリマップ

PC-8801MKIIのメモリマップは次のようになっています。

FFFFH
RAM
64K
バイト
0

VRAM0
16Kバイト

VRAM1
16Kバイト

VRAM2
16Kバイト
FFFFH
C000H

7FFFH
ROM1
32K
バイト
0

ROM2
32K
バイト

ROM3
8Kバイト

オプション
ROM
8Kバイト

オプション
RAM
32Kバイト

メインCPUのメモリマップ

64K RAM
FFFFH
83FFH
1Kバイト
8000H
7FFFH
0

N88-BASICで使われるメモリ

FFFFH
8000H
0

N-BASICで使われるメモリ

7FFFH
RAM
16Kバイト
4000H

7FFH
ROM
0
2Kバイト

サブCPUのメモリマップ

WAIT機能の説明

PC-8801MKⅡではCPUのM1サイクルに自動的に1WAITがかかります。

そのためスロットに増設したROMには，アクセスタイム300ns以内のものを使用することができますが，これよりもアクセスタイムの遅いROMを使用する場合にはCPUのメモリリードサイクルに1 WAITを追加することができます。

次にその方法を示します。

① ディップスイッチ1の7をONにする。

② スロットの$\overline{\text{MWAIT}}$信号をLOWレベルにする。

付録1.4　VRAM

PC-8801MKⅡには，テキストVRAMとグラフィック用VRAMがあります。テキストVRAMには，CRTコントローラが使用され，PC-8001とコンパチブルとなっています。グラフィック用VRAMは，3つの使い方ができます。(640×200ドットカラー，640×200ドット白黒を3ページ，640×400ドット白黒)

また，グラフィック用VRAMとテキストVRAMは同時に表示することができます。

次に画面とグラフィック用VRAMとの対応を示します。

(1)　640×200ドットカラー

グラフィック用VRAMの構成

0 ←ドット→ 639

0 ←ドット→ 7

ドット（縦）0～199

C000	C001	----	C04F
C050	C051		C09F
⋮			
FE30	FE31	----	FE7F

次に，ビットとドットの対応を示します。

	C000		
MSB	7 6 5 4 3 2 1 0	LSB	GVRAM 2 (グリーン)
MSB	7 6 5 4 3 2 1 0	LSB	GVRAM 1 (レッド)
MSB	7 6 5 4 3 2 1 0	LSB	GVRAM 0 (ブルー)

1ドットが1 bitに対応しています。

このモードでは3つの同じ構成のグラフィック用VRAM(GVRAM 0, GVRAM 1, GVRAM 2)があり，それが光の3原色(ブルー，レッド，グリーン)に対しています。次に示す組合せにより，8色使えます。

色＼VRAM	GVRAM 2	GVRAM 1	GVRAM 0
黒	0	0	0
青	0	0	1
赤	0	1	0
紫	0	1	1
緑	1	0	0
水色	1	0	1
黄色	1	1	0
白	1	1	1

(2) 640×200ドット白黒3ページ

このモードは(1)のグラフィック用VRAM構成と同じです。ただし，GVRAM 0，GVRAM 1，GVRAM 2は独立したかたちで使用することにより3ページとなります。

(3) 640×400ドット白黒

グラフィック用VRAM

0 ←ドット→ 639
0 ←ドット→

ドット				
0	C 0 0 0	C001	…	C04F
	C 0 5 0	C051	…	C09F
199	FE30	FE31	…	FE7F
200	C000	C001	…	C04F
	C050	C051	…	C09F
399	FE30	FE31	…	FE7F

次にビットとドットの対応を示します。

C 0 0 0

MSB | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | LSB

(4) テキストVRAM

このテキストVRAMはPC-8001と構造がコンパチブルになっています。表示エリア(画面上の１つのキャラクタと１対１に対応)とアトリビュート(画面の表示を修飾するためのもの)とによって構成されています。

次に画面とテキストVRAMとの対応を示します。

○N_{88}-BASIC(N_{88} DISK-BASIC)

• N-BASIC(PC-8001も同じ)

テキストVRAMエリアの80×25ワードのメモリは，80キャラクタモードの画面に対応しています。

また，40キャラクタモードにした場合はビデオRAMの偶数アドレスが有効となり，1アドレスおきに，画面と対応することになります。(40キャラクタモードにおいて，奇数アドレスのデータは，画面に表示しません。)

アトリビュートエリアは80/40キャラクタモードいずれの場合も有効となります。

36キャラクタモード，72キャラクタモードは，40キャラクタモード，80キャラクターモード時の両側2または4桁を使用していません。ただし，このキャラクタモードは，N-ABSICのみ有効です。

注　意

テキスト画面のカラー指定を行うとき，カラーモード，モノクロモードにかかわらず，1行内で19ヶ所以上変化点を指定するとそれ以上同一の色になったり，同一の表示モードになってしまいます。

付録2　入出力インタフェース

PC-8801MKⅡには，次に示す入出力インタフェースが標準装備されています。

カラーCRTインタフェース
白黒CRTインタフェース
オーディオ・カセットインタフェース
RS-232Cインタフェース
プリンタインタフェース
拡張ミニフロッピィディスク用のインタフェース
汎用I/Oポート
拡張用スロット
キーボード

以下にこれらのインタフェースの入出力規格を述べます。

付録2.1　カラーCRTインタフェース

PC-8801MKⅡは専用カラーディスプレイに接続するためのインタフェースを内蔵しています。このコネクタから出力信号は背面パネルにある8ピンのDINコネクタに出力されています。

次に出力信号について説明します。

カラーCRTインタフェース

端子番号	信　号　名		方向
1	VDD	＋12V	出力
2	GND	信号グランド	
3	COLOR CLK	カラーバースト×4倍のクロック	出力
4	$\overline{\text{HSYNC}}$	水平同期信号	出力
5	$\overline{\text{VSYNC}}$	垂直同期信号	出力
6	R	ビデオ信号(Red)	出力
7	G	ビデオ信号(Green)	出力
8	B	ビデオ信号(Blue)	出力

カラーCRTコネクタピンコネクション

(1) VDD

電源端子で＋12Vが出力されています。

これはオプションのカラーモジュレータに電源を供給するためのものです。

(2) GND

信号グランドで，全ての信号に対してのグランド電位を与えるものです。

(3) COLOR CLK

この信号も，オプションのカラーモジュレータを駆動するためのもので，TTLレベル，14.32MHzのクロックが出力されています。

(4) $\overline{\text{HSYNC}}$

水平同期信号で，TTLレベル，負論理の信号が，出力されています。

(5) $\overline{\text{VSYNC}}$

垂直同期信号で，TTLレベル，負論理の信号が，出力されています。

(6) R

赤色のビデオ信号で，TTLレベル，正論理の信号が，出力されています。

(7) G

緑色のビデオ信号で，TTLレベル，正論理の信号が，出力されています。

(8) B

青色のビデオ信号で，TTLレベル，正論理の信号が，出力されています。

付録２．２　白黒CRTインタフェース

PC-8801MKⅡは白黒モニタテレビに接続するためのインターフェースを内蔵しています。このインタフェースからの出力信号は，背面パネルにある５ピンのDINコネクタに出力されています。

次に出力信号について説明します。

白黒CRTインタフェース

端子番号	信号名		方向
1	VDD	＋12V	出力
2	GND	信号グランド	
3	VIDEO OUT	ビデオ出力信号	出力
4	NC		
5	LPEN	ライトペン信号	入力

NCは，No Connectionの略です。

白黒CRTコネクタピンコネクション

(1)　VDD

電源端子で，＋12Ｖが，出力されています。

(2)　GND

信号グランドで，全ての信号に対してのグランド電位を与えるものです。

(3)　VIDEO OUT

ビデオ信号の出力端子です。この端子から出力されるビデオ信号は，コンポジットビデオ信号で映像信号と複合同期信号とが，すでに合成された信号です。

又，カラーモードで使用した場合，指定した色によって輝度変調が

かかり，映像信号のDCレベルを色に応じて変えることができます。これによって，白黒テレビジョンを使用した場合でも，カラーモードにセットすることによって，画面に濃淡をつけることができます。

(4) LPEN

ライトペンからの入力信号で,TTLレベルの信号が入力されます。

付録2.3 オーディオカセットインタフェース

PC-8801MKⅡは，オーディオカセットへのデータの書き込み及び，オーディオカセットからのデータの読み込みを行うためのインタフェースを内蔵しています。このインタフェースの入出力信号は，背面パネルにある8ピンのDINコネクタに接続されています。

次に，オーディオカセットへの書き込み方式について説明します。

① 書き込み方式

マーク2400Hz，スペース1200HzのFSK方式

② スピード

600bps/1200bps

③ リモートコントロール機能

オーディオカセットテープレコーダについているリモート端子を使用することによって，テープの自動スタート/ストップが可能です。

注 意

オーディオカセットにデータを入出力するコマンドが，終了すると自動的にリモート信号がOFFとなりますので，テープの早送り巻きもどしを行う場合は，リモート端子よりコネクタをぬくかMOTORコマンドにより，リモート信号をONにしてから行って下さい。

次に入出力信号について説明します。

オーディオカセットインタフェース

端子番号	信号名		方向
1	NC		
2	GND	信号グランド	
3	NC		
4	CMTOUT	オーディオカセット書き込み信号	出力
5	CMTIN	オーディオカセット読み込み信号	入力
6	REM＋	リモート	出力
7	REM－	リモート	出力
8	NC		

8 7 6 3 1 5 4 2

オーディオカセットコネクタピンコネクション

(1) GND

信号グランドで，全ての信号に対してのグランド電位を与えるものです。

(2) CMT OUT

PC-8801MKⅡから，オーディオカセットテープレコーダへの出力信号で,オーディオカセットテープレコーダのマイク端子へ接続します。なお，この信号は，付属ケーブルの赤色プラグに出力されています。

(3) CMT IN

オーディオカセットテープレコーダからの入力信号で,オーディオカセットテープレコーダのイヤホーン端子に接続します。

なおこの信号は，付属ケーブルの白色プラグに接続されています。

(4) REM＋．REM－

オーディオカセットテープレコーダのリモート端子を制御する信号で,オーディオカセットテープレコーダのリモート端子に接続します。

この信号は,PC-8801MKⅡ内に内蔵されているリレーの接点信号で,オーディオカセットにデータを入出力するコマンド及び“motor”コマンドより制御されます。

付録2．4　RS-232Cインタフェース

PC-8801MKⅡは，EIA(Electronic Industries Association)の定めたシリアルデータ転送規格のRS-232Cインタフェースを内蔵しています。このインタフェースの入出力信号は背面のパネルにある25ピンのDサブコネクタに接続されています。

このインタフェースはPC-8801MKⅡのターミナルモードにより制御されます。次に，入出力信号について説明します。

RS-232Cインタフェース

端子番号	信号名		方向
1	GND	信号グランド	
2	$\overline{\mathrm{TXD}}$	送信データ	出力
3	$\overline{\mathrm{RXD}}$	受信データ	入力
4	RTS	送信要求	出力
5	CTS	送信可	入力
6	DSR	データセットレディ	入力
7	GND	信号グランド	
8	DCD	キャリア検出信号	入力
9	NC		
10	NC		
11	NC		
12	NC		
13	NC		
14	NC		
15	NC		
16	NC		
17	NC		
18	NC		
19	NC		
20	DTR	データ端末レディ	出力
21	NC		
22	NC		
23	NC		
24	NC		
25	NC		

RS-232Cコネクタピンコネクション

(1) GND

信号グランドで，全ての信号に対してグランド電位を与えるものです。

(2) $\overline{\mathrm{TXD}}$

送信データで，PC-8801MKⅡから外部シリアル入出力機器へ，シリアルデータを出力するためのものです。負論理で出力されています。

(3) $\overline{\mathrm{RXD}}$

受信データで，外部シリアル入出力機器から，PC-8801MKⅡへシリアルデータを入力するためのものです。負論理で入力します。

(4) RTS

送信要求信号で，モデムの送信機能を制御します。正論理で出力されています。

(5) CTS

外部シリアル入出力機器からの入力信号で，外部シリアル入出力機器が，データの送信可能状態にあるか否かをPC-8801MKⅡに示す信号です。

(6) DSR

外部シリアル入出力機器からの入力信号で，外部シリアル入出力機器が，データの送受信ができる状態にあるか否かをPC-8801MKⅡに示す信号です。

(7) DCD

外部シリアル入出力機器からの入力信号で，受信キャリアが，規格内にあるか否かをPC-8801MKⅡに示す信号です。

(8) DTR

外部シリアル入出力機器への出力信号で，PC-8801MKⅡ側の送受信

機能が，レディになったことを外部シリアル入出力機器へ示す信号です。

(9) ボーレートの指定

RS-232Cのボーレートは前面パネルにあるジャンパスイッチにより切り替えることができます。

次にジャパスイッチの位置とボーレートの関係を示します。

ジャンパスイッチとボーレート

付録2.5　プリンタインタフェース

PC-8801MKⅡはプリンタ用パラレルインタフェース(セントロニクス社製プリンタ仕様に準ずる)を内蔵しています。このインタフェースの入出力信号は背面パネルの14ピンアンフェノールコネクタに接続されています。

次に入出力信号について説明します。

プリンタインタフェース

端子番号	信号名
1	$\overline{\text{PSTB}}$
2	PDB0
3	PDB1
4	PDB2
5	PDB3
6	PDB4
7	PDB5
8	PDB6
9	PDB7
10	NC
11	BUSY
12	NC
13	GND
14	GND

7　1
14　8

プリンタコネクタピンコネクション

(1)　$\overline{\text{PSTB}}$

プリンタに出力するライトストローブ信号で，プリンタにデータを

出力する時のライト信号として使います。

(2) PDB0～PDB7

プリンタに出力する8ビットのパラレルデータバスです。

(3) BUSY

プリンタからのレディ信号を受信します。

この信号が，LOWレベルのときPC8801MKⅡは，プリンタにデータを出力することができます。

(4) GND

信号グランドで全ての信号に対してグランド電位を与えるものです。

すべての入出力は，TTLレベルで行います。

付録2.6　汎用I/Oポート

PC-8801MKⅡは汎用のI/Oポートを持っています。このI/Oポートの入出力信号は背面パネルの9ピンのDサブコネクタに接続されています。このポートは入力信号2本，出力信号3本，割り込み1chが接続されています。PC-8801MKⅡのN₈₈-BASICではサポートされておりませんが多様なアプリケーションに対応することができるでしょう。

汎用I/Oポートインタフェース

端子番号	信号名
1	UIP2
2	UIP1
3	UOP1
4	UOP2
5	GND
6	UOP3
7	GND
8	$\overline{\text{UINT2}}$
9	＋5V

1 - - - - - - - - - - 5

6- - - - - - - - -9

汎用I/Oポートコネクタピンコネクション

使用方法

汎用I/Oポートは以下のI/Oアドレスに割りふられています。

7							0	
UOP 2	UOP 1							OUT 40Hポート

7							0	
UOP 3								OUT 10Hポート

7							0	
UIP 2	UIP 1							IN 30Hポート

汎用I/Oポートを使用してデータを出力する場合は，以下の事を注意して下さい。(UOP 1，UOP 2のみ)

☆OUT 40Hポート

N-BASIC使用時は0EA67H，N88-BASIC使用時は0E6C1Hがこのポートのワークエリアになっています。汎用ポートで使用する以外のビットをいじると，動作不良の原因となりますので注意して下さい。

プログラム例(N88-BASIC使用時)

UOP 1に“H”

UOP 2に“L”を出力する場合

```
10  A=&HE6C1              :N-BASICの時は&HEA67。
20  A=(A AND &H6F)        :上位2ビットのみを0にする。
30  A=(A OR &H40)         :上位2ビットを01にする。
40  OUT &H40, A           :下位6ビットはそのままにデー
                           タ出力
50  END
```

付録2．7　拡張用スロットバス

スロット1・2			
サイドA		サイドB	
ピン番号		ピン番号	信号名
1	GND	1	GND
2	GND	2	GND
3	＋5V	3	＋5V
4	＋5V	4	＋5V
5	AB 0	5	
6	AB 1	6	
7	AB 2	7	$\overline{\text{MWAIT}}$
8	AB 3	8	$\overline{\text{INT4}}$
9	AB 4	9	$\overline{\text{INT3}}$
10	AB 5	10	$\overline{\text{INT2}}$
11	AB 6	11	$\overline{\text{FDINT1}}$
12	AB 7	12	$\overline{\text{FDINT2}}$
13	AB 8	13	DB 0
14	AB 9	14	DB 1
15	AB 10	15	DB 2
16	AB 11	16	DB 3
17	AB 12	17	DB 4
18	AB 13	18	DB 5
19	AB 14	19	DB 6
20	AB 15	20	DB 7
21	$\overline{\text{RD}}$	21	$\overline{\text{MEMR}}$
22	$\overline{\text{WR}}$	22	POWER
23	$\overline{\text{MREQ}}$	23	$\overline{\text{IOW}}$
24	$\overline{\text{IORQ}}$	24	$\overline{\text{IOR}}$
25	$\overline{\text{M1}}$	25	$\overline{\text{MEMW}}$
26	$\overline{\text{RAS0}}$	26	$\overline{\text{DMATC}}$
27	$\overline{\text{RAS1}}$	27	DMARDY
28	$\overline{\text{RFSH}}$	28	$\overline{\text{DRQ1}}$
29	$\overline{\text{MUX}}$	29	$\overline{\text{DACK1}}$
30	$\overline{\text{WE}}$	30	4CLK
31	$\overline{\text{ROMKILL}}$	31	$\overline{\text{NMI}}$
32	$\overline{\text{RESET}}$	32	$\overline{\text{WAITRQ}}$
33	SCLK	33	＋12V
34	CLK	34	－12V
35	V1	35	V1
36	V2	36	V2

スロット3			
サイドA		サイドB	
ピン番号	信号名	ピン番号	信号名
1	GND	1	GND
2	GND	2	GND
3	＋5V	3	＋5V
4	＋5V	4	＋5V
5	AB 0	5	
6	AB 1	6	
7	AB 2	7	$\overline{MWAIT}$
8	AB 3	8	$\overline{INT4}$
9	AB 4	9	$\overline{INT3}$
10	AB 5	10	$\overline{INT2}$
11	AB 6	11	$\overline{FDINT1}$
12	AB 7	12	$\overline{FDINT2}$
13	AB 8	13	DB 0
14	AB 9	14	DB 1
15	AB 10	15	DB 2
16	AB 11	16	DB 3
17	AB 12	17	DB 4
18	AB 13	18	DB 5
19	AB 14	19	DB 6
20	AB 15	20	DB 7
21	$\overline{RD}$	21	$\overline{MEMR}$
22	$\overline{WR}$	22	POWER
23	$\overline{MREQ}$	23	$\overline{IOW}$
24	$\overline{IORQ}$	24	$\overline{IOR}$
25	$\overline{M1}$	25	$\overline{MEMW}$
26	$\overline{RAS0}$	26	$\overline{DMATC}$
27	$\overline{RAS1}$	27	DMARDY
28	$\overline{RFSH}$	28	$\overline{DRQ2}$
29	$\overline{MUX}$	29	$\overline{DACK2}$
30	$\overline{WE}$	30	4CLK
31	$\overline{ROMKILL}$	31	$\overline{NMI}$
32	$\overline{RESET}$	32	$\overline{WAITRQ}$
33	SCLK	33	＋12V
34	CLK	34	－12V
35	V1	35	V1
36	V2	36	V2

付録2.8　キーボード

PC-8801MKⅡのキーボードは，ソフトウェアセンス方式をとっています。マトリクス状に配置されたキースイッチは，CPUのインプット命令によりスキャンされ，押されたキーの情報は，ソフトウェアによってJISコードに変換されます。

キーボードのデータを直接見たいときは，BASICのINP関数によって見ることができます。

例

PRINT HEX$（INP（3）） ⏎

を実行して「FF」が表示されたらH～Oのキーは押されていません。もし「FE」が表示されればHが「F 7」が表示されればKが押されていることになります。(下表参照)

$(FF)_{16}=(11111111)_2$　……押されていない
$(FE)_{16}=(11111110)_2$　……Hのキーが押されている
$(FD)_{16}=(11111101)_2$　…… I　〃
$(FB)_{16}=(11111011)_2$　…… J　〃
$(F7)_{16}=(11110111)_2$　……K　〃
$(EF)_{16}=(11101111)_2$　……L　〃
$(DF)_{16}=(11011111)_2$　……M　〃
$(BF)_{16}=(10111111)_2$　……N　〃
$(7F)_{16}=(01111111)_2$　……O　〃

2個のキーが同時に押されているときは上記以外の値になります。例えばIとMが押されていれば，$(11011101)_2=(DD)_{16}$が得られます。

（データ・バス）

（アドレス・バス）

	D0	D1	D2	D3	D4	D5	D6	D7
00	0	1	2	3	4	5	6	7
01	8	9	*	+	=	,	.	⏎
02	@ ゛	A チ	B コ	C ソ	D シ	E イ	F ハ	G キ
03	H ク	I ニ	J マ	K ノ	L リ	M モ	N ミ	O ラ
04	P セ	Q タ	R ス	S ト	T カ	U ナ	V ヒ	W テ
05	X サ	Y ン	Z ッ ツ	[。「	¥	_	] ム 」	^ ヘ	- = ホ
06	0 ヲ ワ	1 ! ヌ	2 ″ フ	3 # ァ ア	4 $ ゥ ワ	5 % ェ エ	6 & ォ オ	7 ▼ ャ ヤ
07	8 (ュ ユ	9) ョ ヨ	: * ケ	; + レ	, < 、 ネ	. > 。 ル	/ ? ・ メ	ー ロ
08	HOME CLR	↑	→	INS DEL	GRPH	カナ	SHIFT	CTRL
09	STOP	f・1	f・2	f・3	f・4	f・5	SPACE	ESC
0A	HTAB	↓	←	HELP	COPY	—	／	CAPS
0B	ROLL UP	ROLL DOWN						

付録3　ディスクユーティリティの使い方

付録3.1　概　　要

N88DISK-BASICシステムディスクには，N88DISK-BASICおよびDISK-BASICのためのユーティリティプログラムが入っています。このディスクユーティリティは，システムディスクを作ったり，フロッピィディスクのコピーや，ファイル転送を行ったりするものです。

• **フロッピィディスクのコピー**

フロッピィディスクにセーブされたファイルは壊れることがあります。例えば，まちがって必要なファイルをkill文で消してしまったり，DSKO$命令でファイルの内容を壊したり，また，ノイズなどハードウェアのトラブルなどで壊れる可能性もあります。

そこで，重要な情報が記録されているフロッピィディスクは，定期的にコピーを作っておいて，もしファイルが壊れても，そのコピーファイルを用いてその被害を最小限にすることが必要です。このようなコピーを行なうために「**COPY DISK**」という機能があります。

「**COPY DISK**」は，ミニフロッピィディスクはミニフロッピィディスクに，標準フロッピィディスクは標準フロッピィディスクにコピーするものです。ミニフロッピィディスクを標準フロッピィディスクにコピーしたり，その逆を行うことはできません。

• **システムディスクの作成**

N88DISK-BASICやDISK-BASICを使うときには，システムディスクが必要です。システムディスクを作るには，「**CREATE SYSTEM DISK**」という機能を使います。これは，N88DISK-BASIC用とDISK-BASIC用に分かれていますので，必要に応じて使い分けてください。

PC-8801MKⅡ(model 20, model 30)には，N88DISK-BASICのシステムディスクが1枚添付されていますが，これをもとにして，新たにシステムディスクを作ることができます。

なお，DISK-BASICのシステムディスクは，添付されていませんので，最初は別のユーティリティプログラム(“ngen. n88”)を使って作ります。以後は，それをもとにして作っていきます。

・データディスクの作成

買ったばかりのフロッピィディスクには，プログラムやデータをセーブすることはできません。未使用のフロッピィディスクを読み書き可能にする機能が「**CREATE DATA DISK**」です。

これによって処理されたフロッピィディスクは，プログラムやデータを読み書きできる状態になります。

システムディスクは，データディスクにN88DISK-BASICやDISK-BASICのディスクコードが書き込まれたものです。したがって，システムディスクにもプログラムやデータのセーブが可能です。

・ファイル転送

「**COPY DISK**」では，同じ種類のフロッピィディスク間のコピーしかできません。それに対して，異なった種類のフロッピィディスク間のファイルを転送する機能が「**TRANSFER FILES**」です。

「**TRANSFER FILES**」は，片面倍密度ミニフロッピィディスク，両面倍密度ミニフロッピィディスクおよび標準フロッピィディスクの間で，どの方向にでも，全ファイルをコピーします。

・IDセクタの書き替え

IDセクタを書き替えることにより，N88DISK-BASICおよびDISK-BASICをスタートさせると同時に，ファイル数をセットし，任意のプログラムを自動的に実行できます。

「**SET UP INFO**」は，このIDセクタを書き替えるものです。

N88DISK-BASICとDISK-BASICでは，IDセクタの内容が一部異なりますので，「**SET UP INFO**」には2つのものが用意されています。

ディスクユーティリティは，N88DISK-BASICモードで実行します。DISK-BASICモードで使うことはできません。また，拡張命令パッケージは切り離した状態でお使いください。

※物理フォーマットとシステムフォーマット

(1) 物理フォーマット

「12.7ディスクファイルの構造」で述べたサーフェス，トラック，セクタといったアドレスをフロッピィディスクにつけることを物理

フォーマットといいます。

未使用のミニフロッピィディスクは，必ず物理フォーマットを行わなければなりません。

N88DISK-BASIC用にNECが供給している標準フロッピィディスク（PC-8886）は，物理フォーマット済みですので，その必要はありません。

物理フォーマットを行なうと，DSKI＄，DSKO＄命令を使えるようになりますが，他のコマンド，ステートメントを使用することはできません。

(2) システムフォーマット

システムフォーマットを行うと，FAT，ディレクトリ，IDの初期化が行われます。これで，そのフロッピィディスクに対して，BASICのコマンド，ステートメントを実行できるようになります。

付録3.2　ディスクユーティリティプログラムとメニューの選択

ディスクユーティリティプログラムには2つのプログラム("dskut 2", "dskut 1")が用意されており，ミニフロッピィディスクの数によって使いわけます。

ミニフロッピィディスクが2ドライブ以上ある場合は，"dskut 2"を使えばよいのですが，1ドライブの場合は内容によって，"dskut 2"と"dskut 1"を選択して使います。

ディスクユーティリティプログラムとメニューの選択を表にまとめておきます。

〈表の見方〉

機能			
フロッピィディスクのタイプ [ミニ] …5インチミニフロッピィディスク [標準] …8インチ標準フロッピィディスク	(システムディスクのタイプ)	ディスクユーティリティプログラム名	メニューの名前

システムディスクのタイプ

[N] ……DISK-BASIC

[N88] ……N88DISK-BASIC

(1)　ミニフロッピィディスクが1ドライブの場合

フロッピィディスクのコピー			
[ミニ] → [ミニ]	[N] [N88]	dskut1	COPY DISK
[標準] → [標準]	[N88]	dskut2	COPY DISK

N88DISK-BASICのシステムディスクを作る			
ミニ *→ ミニ	N88	dskut1	CREATE SYSTEM DISK [N88]
ミニ *→ 標準 標準 *→ ミニ 標準 *→ 標準	N88	dskut2	CREATE SYSTEM DISK [N88]

＊N88DISK-BASICシステムディスク

DISK-BASICのシステムディスクを作る			
ミニ *→ ミニ	N	dskut1	CREATE SYSTEM DISK [N]

＊DISK-BASICシステムディスク

データディスクを作る			
ミニ	N N88	dskut2	CREATE DATA DISK
標準	N88		

ファイルの転送			
ミニ → 標準 標準 → ミニ	N88	dskut2	TRANSFER FILES

N88DISK-BASICシステムディスクのIDセクタの書き換え			
ミニ 標準	N88	dskut2	SET UP INFO [N88]

DISK-BASICシステムディスクのIDセクタの書き換え			
ミニ	N	dskut2	SET UP INFO [N]

(2) ミニフロッピィディスクが2ドライブの場合

フロッピィディスクのコピー			
ミニ → ミニ	N N88	dskut2	COPY DISK
標準 → 標準	N88		

N88DISK-BASICのシステムディスクを作る			
ミニ *→ ミニ ミニ *→ 標準 標準 *→ ミニ 標準 *→ 標準	N88	dskut2	CREATE SYSTEM DISK [N88]

＊N88DISK-BASICシステムディスク

DISK-BASICのシステムディスクを作る			
ミニ *→ ミニ	N	dskut2	CREATE SYSTEM DISK [N]

＊DISK-BASICシステムディスク

データディスクを作る			
ミニ	N N88	dskut2	CREATE DATA DISK
標準	N88		

ファイルの転送			
ミニ → ミニ	N N88	dskut2	TRANSFER FILES
ミニ → 標準 標準 → ミニ	N88		

N88DISK-BASICシステムディスクのIDセクタの書き換え			
ミニ 標準	N88	dskut2	SET UP INFO [N88]

DISK-BASICシステムディスクのIDセクタの書き換え			
ミニ	N	dskut2	SET UP INFO [N]

付録3.3 ディスクユーティリティの起動

“dskut 1”, “dskut 2”ともに同じように使います。

(1) N88DISK-BASICをスタートさせます。

(注 意)

拡張命令が追加されている場合には，ディスクユーティリティは使えません。cmd cut文で拡張命令を切り離してから使ってください。

(2) ディスクユーティリティプログラムをロードします。

load “<デバイス名> dskut1. n88” ⏎

load “<デバイス名> dskut2. n88” ⏎

<デバイス名> は，ディスクユーティリティが入っているドライブのデバイス名です。通常は“1:”ですから省略できます。

(3) プログラムを実行させます。

run ⏎

(4) しばらくすると，メニュー画面になりますので，必要な処理を選択して使ってください。

具体的な使い方については，次節で述べます。

付録3．4　“dskut2”の使い方

“dskut 2”のメニュー画面は，次のようになります。

```
DISK UTILITY for PC-8801MK2 dskut2
           MENU
COPY DISK
CREATE SYSTEM DISK  [N88]
CREATE SYSTEM DISK  [N]
CREATE DATA DISK
TRANSFER FILES
SET UP INFO         [N88]
SET UP INFO         [N]
Select a feature and press RETURN key.
```

ここでは，カーソル移動キー(↑，↓)を使って，実行したい処理を選択し，⏎ キーを押します。

次に，各機能について説明します。もし誤った操作をしてしまったら，STOP キーを押して，もう一度プログラムを実行してください。

COPY DISK(同じ種類のフロッピィディスクのコピー)

(1) 「COPY DISK」を選択すると次のメッセージが現われます。

[[[COPY DISK]]]

Mount MASTER disk on drive.

Drive number?

コピーしたいフロッピィディスク(マスターディスク)をドライブにセットし，そのドライブのデバイス番号を入力します。デバイス番名“1:”のドライブにセットした場合は，

1⏎

と入力します。

(2) 次に新しいディスクをドライブにセットするように指示してきます。

Mount NEW disk on drive.

Drive number?

新しいフロッピィディスク(または，すべて消してもかまわないフロッピィディスク)をセットし，そのドライブに割り当てられているデバイス番号を入力します。

このとき，マスターディスクと新しいフロッピィディスクの種類が合わない場合は，

Can't copy.

と表示され，(1)に戻ります。

(3) 次に物理フォーマットをするかどうか聞いてきます。

Do you need physical formatting (y/n)?

物理フォーマットをする必要がある場合(特に，新しいミニフロッピィディスク)には，ここで，

y [↵]

と入力します。nを入力すると(4)に移ります。

すると，

Physical format a disk on drive X.

Sure (y/n)?

と表示して，物理フォーマットの実行を確認してきます。

Xは，(2)で入力した値です。

y [↵]

と入力すると，

Physical formatting.

と表示して，物理フォーマットを行います。

さらに，標準フロッピィディスクのフォーマットを行うときは，

Formatting track XX

⋮

と表示されます。

(4) 物理フォーマットが終わると，コピーが始まります。

マスターディスクの入っているドライブと，新しいフロッピィディ

スクの入っているドライブのアクセス表示用LEDが順に点灯すると同時に，画面にコピーしているトラック番号が表示されます。

Copying track XX

⋮

（“XX”はとびとびの値のこともあります。）

(5) 無事に終了すれば，

Completed.

と表示されます。

CREATE SYSTEM DISK ［N88］（N88DISK-BASICのシステムディスクを作る）

(1) 「CREATE SYSTEM DISK ［N88］」を選択すると次のメッセージが現われます。

[[[CREATE SYSTEM DISK]]]

N88DISK-BASIC

Mount SYSTEM disk on drive.

Drive number?

N88DISK-BASICのシステムディスクをドライブにセットし，そのドライブのデバイス番号を入力します。

(2) 次に

Mount NEW disk on drive.

Drive number?

と表示されますので，新しいフロッピィディスクをドライブにセットし，そのドライブのデバイス番号を入力します。

システムディスクと新しいフロッピィディスクの種類は異なってもかまいません。

(3) 次に，物理フォーマットをするかどうか聞いてきます。

Do you need physical formatting (y/n)?

物理フォーマットの必要がある場合は，

y ⏎

と入力します。nを入力すると(4)に移ります。

すると，

Physical format a disk on drive X.

Sure (y/n)?

と表示して，実行を確認してきます。

y [⏎]

と入力すると，

Physical formatting.

と表示され，物理フォーマットが行われます。さらに，標準フロッピィディスクの場合は，

Formatting track XX

⋮

と表示されます。

(4) 物理フォーマットが終わると，システムフォーマットが行われさらに，

Copying system.

と表示されて，システムのコピーが行われます。

(5) 次のようにして表示されれば終わりです。

Completed.

CREATE SYSTEM DISK [N] (DISK-BASICのシステムディスクを作る)

使い方は，N88DISK-BASICシステムディスクを作るときと同様です。

ただし，DISK-BASICでは標準フロッピィディスクは使えませんので，ドライブのデバイス番号に標準フロッピィディスクドライブを指定することはできません。

CREATE DATA DISK(データディスクを作る)

(1) 「CREATE DATA DISK」を選択すると，次のメッセージが現われます。

[[[**CREATE DATA DISK**]]]

Mount NEW disk on drive.

Drive number?

新しいフロッピィディスクをドライブにセットし，そのドライブのデバイス番号を入力します。

(2) 次に，物理フォーマットをするかどうか聞いてきます。

Do you need physical formatting (y/n)?

物理フォーマットに必要な場合は，

y⏎

と入力します。nを入力すると，(3)へ移ります。

すると，

Physical format a disk on drive X.

Sure（y/n）

と確認してきますので，実行する場合は，

y⏎

と入力します。

Physical formatting.

と表示され，物理フォーマットが行われます。

さらに，標準フロッピィディスクの場合は，

Formatting track XX.

⋮

と表示されます。

(3) 次にシステムフォーマットが行われ，

Completed.

と表示されて，終了します。

TRANSFER FILES（ファイル転送）

(1) 「TRANSFER FILES」を選択すると次のメッセージが現われます。

[[[**TRANSFER FILES**]]]

Transfer from?

マスターディスクをドライブにセットし，そのデバイス番号を入力します。さらに，マスターディスクが片面倍密度ミニフロッピィディスク（PC-8031，PC-8031-1W）で使用していたフロッピィディスクの場合は，数字の後に“S”を付けます。

デバイス名が「1：」で，両面倍密度ミニフロッピィディスクの場合は，

1⏎

デバイス名が「2：」で，片面倍密度ミニフロッピィディスクの場合は

2 s ⏎

と入力します。

(2) 正しく入力されると，

to?

と表示されますので，転送されるフロッピィディスクをドライブにセットし，そのドライブのデバイス番号を入力します。　この場合も，片面倍密度ミニフロッピィディスクを指定することができます。

なお，このとき転送先のフロッピィディスクは，すでにフォーマットされていなければなりません。(「CREATE SYSTEM DISK」または「CREATE DATA DISK」を使います。)

(3) 次に，

Sure (y/n)?

と確認してきます。もし，今まで指定したドライブのデバイス番号がまちがっていれば，

n ⏎

と入力します。

y ⏎

を入力すると，

Copying file XXXX　X→X

と表示して，ファイルを転送します。

(4) すべてのファイルが転送し終わると，

Completed.

と表示されて，終了します。

注　意

もし，転送されるフロッピィディスクの未使用セクタ数より，マスターディスクが使っているセクタ数の方が大きい場合は，途中で

Disk full

というエラーメッセージが表示されます。このプログラムを実行する前に，転送できるか確認するようにしてください。

また，転送されるフロッピィディスクに同じファイル名のものがあるとそのファイルは転送するファイルの内容に置き替わります

ので注意して下さい。

SET UP INFO［N88］（N88DISK-BASICのIDセクタの書き換え）

(1)「SET UP INFO [N88]」を選択すると，次のメッセージが現われます。

[[[SET UP INFO]]]

N88 DISK-BASIC

Mount SYSTEM disk on drive.

Drive number?

IDセクタを書き換えたいN88DISK-BASICシステムディスクをドライブにセットし，そのドライブのデバイス番号を入力します。

(2) 次からは，IDセクタに書き込む内容を設定していきます。

まず，

How many files(−1 or 0−15)?

とたずねてくるので，同時にオープンしたいファイル数を入力します。例えば，3つオープンさせたい場合ですと，

3 ⏎

と入力します。

このとき，−1を入力すると，IDのファイル数には255(16進でFF)がセットされますから，ファイル数はセットされず，スタートするたびにファイル数を尋ねるようになります。

(3) 次に，

Year for init(- 1 or 0 -99)?

と表示されますので，スタート時にセットするカレンダ時計の年を入力します。セットする必要がない場合は−1を入力して下さい。もし，セットしておかないと，スタート時の年は84に固定されます。

注　意

(2)および(3)で，⏎ キーのみ入力すると，どちらも0がセットされたことになります。

(4) さらに，

Text for init?

と表示されますから，N88DISK-BASICのスタートと同時に実行したいBASICのテキスト(コマンドやステートメント)を入力します。例えば，

“test. n88”というプログラムをスタートさせたい場合ですと,

run ”test.n88” [⏎]

と入力します。

もし，ここで入力したBASICのテキストが長すぎた場合は,

Text can have a maximum of XXX characters

と表示してブザーを鳴らし，もう一度，BASICのテキストの入力を行うように指示してきます。

(5) 以上の入力が終わると，確認のために，今までセットした内容が表示されます。

- オープンするファイルの数をセットしない場合

  Number of files is asked when starting up

- オープンするファイルの数をセットした場合

  Number of files to allocate is XX

- 年およびBASICのテキストをセットした場合

  Year for init is XX

  Text for init is XXXXX

(6) 次に,

Sure (y/n)?

と聞いてきますので，正しければ

y [⏎]

と入力します。

n [⏎]

のときは，(2)へ戻ります。

(7) 次のように表示されて終了します。

Completed.

SET UP INFO [N] (DISK-BASICのIDセクタの書き換え)

使い方は,「SET UP INFO [N88]」と同じです。ただし，標準フロッピィディスクドライブを指定することはできません。

付録3.5 “dskut1”の使い方

”dskut 1 ”のメニュー画面は，次のようになります。

DISK UTILITY for PC-8801MK2 dskut1

MENU
COPY DISK
CREATE SYSTEM DISK [N88]
CREATE SYSTEM DISK [N]

Select a feature and press RETURN key.

ここでは，カーソル移動キー(↑，↓)を使って，実行したい処理を選択し，⏎ キーを押します。

次に，各機能について説明していきます。

COPY DISK(ミニフロッピィディスクのコピー：1ドライブ)

(1) 「COPY DISK」を選択すると，次のメッセージが現われます。

[[[COPY DISK]]] (5 ” 1 drive)

Do you need physical formatting (y/n)?

新しいミニフロッピィディスクの物理フォーマットが必要な場合は，

y ⏎

と入力します。そうでなければ，

n ⏎

と入力します。この場合は，(3)に移ります。

(2) 物理フォーマットを行なう場合は，

Mount NEW disk on drive X.

Sure (y/n)?

と表示されますので，新しいミニフロッピィディスクをデバイス番号Xのドライブにセットし，

y ⏎

と入力してください。

Physical formatting.

と表示されて，物理フォーマットが行われます。

(3) 次に，すべてのファイルがコピーされるまで，以下の操作を繰り返します。

(4) 次のように表示されたら，マスターディスクを入れます。

Mount MASTER disk on drive X.

Sure (y/n)?

正しくセットできたら，

y ⏎

を入力してください。

Reading track...

と表示されながら，データが読み込まれます。

(5) 次に，

Mount NEW disk on drive X.

Sure (y/n)?

と表示されたら，マスターディスクを取り出して，新しいミニフロッピィディスクを入れます。

正しくセットできたら，

y ⏎

を入力して下さい。

Writing track...

表示されながら，データが書き込まれます。

ここで，すべてのファイルがコピーされると，

Completed.

と表示され終了します。

そうでない場合は，(4)へ戻ります。

CREATE SYSTEM DISK［N88］（N88DISK-BASICのシステムディスクを作る：１ドライブ）

(1)「CREATE SYSTEM DISK［N88］」を選択すると，次のメッセージが現われます。

　　[[[CREATE SYSTEM DISK]]]（5"1 drive）

　　　　N88DISK-BASIC

　　Do you need physical formatting (y/n)?

新しいミニフロッピィディスクの物理フォーマットが必要な場合は，

　　y ⏎

と入力します。そうでなければ，

　　n ⏎

と入力します。この場合は，(3)に移ります。

(2) 物理フォーマットを行う場合は，

　　Mount NEW disk on drive X

　　Sure (y/n)?

と表示されますので，新しいミニフロッピィディスクをデバイス番号Xのドライブにセットし，

　　y ⏎

と入力してください。

　　Physical formatting.

と表示されて，物理フォーマットが行われます。

(3) 次に，

　　Mount SYSTEM disk on drive X.

　　Sure (y/n)?

と表示されますので，N88DISK-BASICシステムディスクをセットし，

　　y ⏎

と入力します。

　　Reading disk code.

と表示され，ディスクコードが読み込まれます。

(4) ディスクコードが読み込まれると，次のように表示されます。

　　Mount NEW disk on drive X.

Sure (y/n)?

ここで，システムディスクをとり出し，物理フォーマットが終わったミニフロッピィディスクをセットします。

正しくセットしたら，

y [⏎]

と入力します。

Writing disk code.

Completed.

と表示されれば終了です。

CREATE SYSTEM DISK [N]（DISK-BASICのシステムディスクを作る：1ドライブ）

使い方は，N88DISK-BASICシステムディスクを作るときと同様です。

付録４　DISK-BASIC

付録４．１　DISK-BASICシステムディスク

N-BASICモードでディスク版BASICを使うには，「DISK-BASICシステムディスク」が必要です。(「第５章システムの起動」参照)

「DISK-BASICシステムディスク」は，本体に添付されていませんが，「N88DISK-BASICシステムディスク」の中にあるユーティリティプログラム“ngen. n88”で作ることができます。

次に，その手順を説明します。

(1) N88DISK-BASICを起動します。(標準フロッピィディスクユニットは切り離しておいてください)

(2) 「N88DISK-BASICシステムディスク」から“ngen. n88”をロードします。

load “ngen. n88” [↵]

(3) プログラムを実行します。

run [↵]

(4) 次のメッセージが画面に表示されますので，「N88DISK-BASICシステムディスク」をとり出し，新しいミニフロッピィディスクをドライブ１にセットします。

DISK-BASIC SYSTEM DISK GENERATOR
Mount NEW disk on drive 1.
Sure (y/n)?

(5) セットしたミニフロッピィディスクを確認したら，

y [↵]

と入力してください。

(6) 次のように表示されながら，システムディスクが作られます。

Physical formatting.
Writing track X sector X.
⋮

(7) しばらくたって，ブザーがなり，次のように表示されれば終了です。

Completed.

(8) 以後，新しい「DISK-BASICシステムディスク」を作るときは，こ

れをもとにして，ディスクユーティリティ“dskut 2”または，“dskut 1”を使います。

注　意

DISK-BASICでは，PC-8881などの標準フロッピィディスクユニットを使うことはできません。

付録4．2　フロッピィディスクの使用宣言と使用終了の宣言

DISK-BASICを使うときに忘れてはならない大切な注意事項として，フロッピィディスクの使用宣言と使用終了の宣言があります。(N88DISK-BASICでは必要ありません)

フロッピィディスクを使用するときには，まず最初にDISK-BASICに対して，これからフロッピィディスクを使用する旨，伝えておくことが要求されます。ことわりもなくフロッピィディスクをアクセスするコマンドを使用するとDISK-BASICはへそをまげて，

Disk not mounted

という怒りのエラーメッセージを返してきます。

フロッピィディスクの使用宣言をするのには次のコマンドを使います。

このとき，指定したドライブには，あらかじめこれから使用するフロッピィディスクをいれておきます。

mount＜ドライブ番号＞

たとえば，ドライブ1を使うのなら，

mount 1

とします。すべてのドライブを使うには，＜ドライブ＞を省略して，単に

mount

とすればよいでしょう。

mountコマンドを入力すると，「カチン」という音がして，LEDがつき，すぐにLEDが消え，Okが表示されます。今後，この「カチン」という音がして，ディスクに対して読み書きすることを，「ディスクをアクセスする」と言うことにします。LEDがついているということは現在ディスクをアクセス中であることを示しています。

話をもとに戻して，mountコマンドを入力すると指定したドライブのディスクをアクセスします。このとき，実際にはFAT(エフエイティーまたはファット)とよばれる一種の住所録をディスクから読んでメモリ内に書き込むのです。

すべてのファイルは名前がついており(ファイル名)，BASICは各ファイルの名前と，そのファイルが実際に格納されているディスク上の位置(住所)とを関連づける住所録を持っています。あなたが，たとえばABCという名前のファイルを読み込むように指令した場合，BASICはこの住所録からABCというファイルの住所を調べて，そのファイルを読みます。

とにかく，mountコマンドを実行すると，そのフロッピィディスクが使用可能になります。

フロッピィディスクの使用を開始するには，mountコマンドによりDISK-BASICに対しその旨の宣言を行います。同様にそのフロッピィディスクの使用を終えるときには，やはりその旨をDISK-BASICに知らせてやる必要があります。これがremoveコマンドです。

remove＜ドライブ番号＞

例

```
remove 1
remove 1， 2
remove
```

最後の例はすべてのディスクの使用終了を宣言します。

以上の説明のうち，「同様にそのフロッピィディスクの使用を終えるとき」という文の持つ意味を十分に理解してください。「そのドライブの」ではありません。

たとえばドライブ1にあるフロッピィディスクをいれてmountを実行し，load，save等を行いました。次に同じくドライブ1に別のディスクをいれて使おうという時，ディスクをいれかえる前にremove 1を実行し，新しいディスクをいれた後にmount 1を実行することが要求されているのです。

removeコマンドを実行すると指定したドライブのディスクがアクセスされます。これはFAT(住所録)をディスクへ書き込むのです。mountを実行するとディスクのFATがメモリ内に読み込まれます。それ以後，FATの更

新は通常メモリ内のFATのコピーに対して行われます。すなわち，最新のFATはメモリ内にあります。removeを実行すると，その最新のFATをディスクに書き込み，ディスク内のFATを正しく更新するのです。

おそらくこの説明でremoveコマンドの重要性が理解できたでしょう。

removeせずにディスクをとりだすと，そのディスクのFATは古いままです。ところが，実際のデータやプログラムは変化してしまっています。つまりFATとデータにくい違いが生じるのです。例えてみれば，住民の転出入があったのに，住所録が古いままになっているのと同じです。BASICにとっては，ファイルの住所を知るにはFATだけが頼りなのであることを忘れてはなりません。こうなってしまうと，そのディスクは壊れたも同然であり，最悪の場合，二度と使えません。すべてのファイル(データ，プログラム)が失なわれるのです。

このディスクは物理的に破壊されたわけではないからフォーマットをやりなおせば再使用はできます。しかし失なわれたデータは二度ともとには戻りません。大事なのはディスクそれ自身ではなくそこに記録されているデータなのです。

付録5　添付ソフトウェアの使い方

付録5.1　添付ミニフロッピィディスク

PC-8801MKⅡ model 20, model 30には，2枚のミニフロッピィディスクが添付されています。

(1)　N88DISK-BASICシステムディスク

N88DISK-BASICを起動するときに使います。この中には，ユーティリティを始め,デモ,チューターなどのすぐに役立つプログラムが入っています。

• ngen. n88

DISK-BASICのシステムディスクを作るためのユーティリティプログラムです。使い方は，「付録4.1 DISK-BASICシステムディスク」を参照してください。

• dskut 1.n88　dskut 2.n88

フロッピィディスクのコピーや，ファイル転送などを行うユーティリティプログラムです。使い方は「付録3 ディスクユーティリティの使い方」を参照してください。

• demo. n88

PC-8801MKⅡのデモプログラムです。PC-8801MKⅡのグラフィック機能を生かしたアニメーションや音楽をお楽しみください。

[実行の方法]

このミニフロッピィディスクをドライブ1に入れ

run "demo. n88" ⏎

とします。

• tutor. n88

パソコン家庭教師プログラムです。キーボードの使い方やグラフィックスの概念などが学べます。

[実行の方法]

このミニフロッピィディスクをドライブ1に入れ，

run "tutor. n88" ⏎

とします。

• **@load. n88　@exst＊**

拡張命令パッケージです。使い方は，「第15章拡張命令」を参照してください。

• **demoex. n88**

拡張命令を使ったサンプルプログラムです。

[実行の方法]

まず，拡張命令を追加しておきます。

run "@load. n88"

すでに拡張命令が追加されている場合は必要ありません。

次に，

run "demoex. n88"

で実行します。

このフロッピィディスクには，他にもいくつかファイルが含まれていますが，これらは，demoおよびtutorで使用するファイルです。

（注　意）

demo，tutorは，標準フロッピィディスクが接続されている場合は正常に動作しません。

(2)　ゲームディスク

「ジャングルアドベンチャー」，「ザ・マウス」の２本のゲームプログラムが入っています。

[実行の方法]

このミニフロッピィディスクをドライブ１に入れ，リセットスイッチを押してください。自動的にスタートします。

なお，標準フロッピィディスクユニットが接続されている場合は，電源を「OFF」にしておいてください。

初期画面が出終ったら，ファンクションキー［f･1］または［f･5］でゲームを選択してください。

• **ジャングルアドベンチャー**

実行後，何かキーを押していくことにより説明が出ます。ゲームが始まりましたら，2, 4, 6, 8のキーを押し，迷路のようなジャングルの中を歩いて2人の女の子を人間の世界へつれてもどってく

ださい。

・ザ・マウス

ねずみがチーズを自分の巣へ持ちかえるゲームです。ねずみをねらっているネコやタカなどに注意して高得点をねらってください。

注　意

5インチのシステムディスクから，ディスクユーティリティ「TRANSFER FILES」によって，8インチや5インチの他のメディアに転送した場合，demo. n88, tutor. n88, ジャングルアドベンチャー，ザ・マウスは正常に動きません。

付録5.2　添付カセットテープ

PC-8801MKII model 10には，カセットテープが添付されています。この中にはデモ，ゲーム，拡張命令パッケージなどが含まれています。

A面：（1） demo…デモプログラム

load "cas2 : demo" ⏎

でロードし，

run ⏎

で実行します。

（2） game…ゲームプログラム

load "cas2 : game" ⏎

でロードし，

run ⏎

で実行します。

B面：（1） @ exst…拡張命令パッケージ

使い方は，「第15章拡張命令」を参照してください。

（2） demoex…拡張命令デモプログラム，

拡張命令を付加した後，

load "cas2 : demoex" ⏎

でロードし，

run ⏎

で実行します。

付録6　PCシリーズの周辺装置

PC-8801MKⅡは豊富なPCシリーズの周辺装置のほとんどを接続する事ができます。次に，PC-2000，PC-6000，PC-8000，PC-8200，PC-8800シリーズの周辺装置について説明します。

PC-8801MKⅡに接続して使用できるもの

[カラーディスプレイ]

R.G.B.セパレート方式のカラーディスプレイを接続する事ができます。次に示すカラーディスプレイは，水平走査周波数が，24.83kHzの専用高解像度ディスプレイで最大640×400ドットのグラフィックが表示できます。

型名	品名	特長
PC-KD551	14インチ・カラーディスプレイ（高解像度）	640×400ドットのグラフィックが表示可能な14型専用高解像度カラーディスプレイ。 ドットピッチ0.39mm，前面フィルタ付き。 （接続ケーブル添付）
PC-8853N	14インチ・カラーディスプレイ（高解像度）	640×400ドットのグラフィックが表示可能な14型専用高解像度カラーディスプレイ。 ドットピッチ0.31mm，前面フィルタ付き。 （接続ケーブル添付）

[モノクロディスプレイ]

コンポジットビデオ方式のモノクロディスプレイを接続する事ができます。次に示すモノクロディスプレイは，水平走査周波数が24.83kHzの専用高解像度ディスプレイで最大640×400ドットのグラフィッが表示できます。

型名	品名	特長
PC-8841	12インチ・モノクロディスプレイ（高解像度）	640×400ドットのグラフィックが表示可能な12型専用高解像度モノクロディスプレイ。（接続ケーブルは本体に添付）
PC-8851	14インチ・モノクロディスプレイ（高解像度）	640×400ドットのグラフィックが表示可能な14型専用高解像度モノクロディスプレイ。（接続ケーブルは本体に添付）

[データレコーダ]

プログラムやデータのロードやセーブは家庭用のテープレコーダを使用して充分行えますが，PC-8801MKⅡ用に新しく購入される場合はPC-2000シリーズ，PC-DRシリーズの専用データレコーダをお勧めします。一般のテープレコーダでは，イヤホンジャック端子にジャックを差し込んでしまうとスピーカから音がでませんし，リモート端子を使用していると巻きもどしや早送りができませんが専用データレコーダでは，この様な点も十分考慮されており，使い易くなっています。

型名	品名	特長
PC-2081	データレコーダ	電池駆動可能なポータブルタイプ モニタ用圧電ブザー内蔵
PC-DR311	データレコーダ	縦型の新デザイン モニタ用圧電ブザー内蔵
PC-DR321	データレコーダ	モニタスピーカ内蔵，モニタレベル独立調整可能，前後5つまでのプログラムサーチ機能付，操作はソフトタッチオペレーション
PC-DR330	データレコーダ	モニタスピーカ内蔵,モニタレベル独立調整可能,プログラム頭出し可,操作はソフトタッチオペレーション

（接続ケーブルは本体添付）

[ミニフロッピィディスク]

PC-8801MKⅡは本体内に最大2ドライブまでのミニフロッピィディスクを内蔵することができます。また，それ以上増設したい場合には，拡張用ミニフロッピィディスクユニットを接続する事により最大4ドライブまで拡張する事ができます。

型名	品名	特長
PC-8801MKII -FD1	増設用ドライブユニット	本体内増設用ミニフロッピィディスクドライブ。(両面倍密度)
PC-80S32	拡張用ミニフロッピィディスクユニット	両面倍密度ミニフロッピィディスクドライブ2台実装の拡張用ユニット。本体内に2ドライブ実装後の拡張に使用。(接続ケーブル添付)

注意

PC-8801MKIIにはPC-8031-1V/1W/2W，PC-8032-1W/2W，PC-80S31は接続できません。

[8インチフロッピィディスク]

PC-8801MKIIは最大4ドライブまでの8インチフロッピィディスクドライブを接続する事ができます。この場合PC-8801MKIIの拡張スロットにインタフェースボードを差して使用します。

型名	品名	特長
PC-8881	8インチ・フロッピィディスクユニット	8インチ薄型ドライブ2台実装，DMA転送方式による高速データ転送 容量1Mバイト/ドライブ 本体の左側設置用 (PC-8884添付 インタフェースボード添付 接続ケーブル添付)
PC-8881(1)	8インチ・フロッピィディスクユニット	PC-8881と同機能 本体の右側設置用
PC-8882	拡張用8インチフロッピィディスクユニット	8インチ薄型ドライブ2台実装， PC-8881の拡張用ユニット 本体の左側設置用 (接続ケーブル添付)
PC-8882(1)	拡張用8インチフロッピィディスクユニット	PC-8882と同機能 本体の右側設置用

[システムディスク]

model 10を購入されたお客様がフロッピィディスクを増設された場合にはシステムディスクが必要になります。

型　名	品　　名	特　　　　長
PC-8837-2W	PC-8801MKII用 システムディスク	両面倍密度ミニフロッピィディスク用
PC-8887	PC-8801MKII用 システムディスク	8インチ・フロッピィディスク用

[ハードディスク]

PC-8801MKIIにはインタフェースボードを拡張スロットに挿入する事によりPC-9800シリーズの5インチ固定ディスクユニットを接続する事ができます。

注　意

PC-8801MKIIのN88-BASIC，N-BASICでは固定ディスクを使用する事ができません。CP/Mや特定アプリケーションのもとでお使いください。

型　名	品　名	特　　　　長
PC-8801-07	5インチ・固定ディスクインタフェースボード	本体と5インチ・固定ディスクユニットを接続用 本体の拡張スロットに実装
PC-98H31	5インチ・固定ディスクユニット	1ユニット目用，容量5Mバイト DMA転送方式による高速データ転送 (インタフェースボードとの接続ケーブル添付)
PC-98H32	拡張用5インチ固定ディスクユニット	2ユニット目用，容量5Mバイト DMA転送方式による高速データ転送 (接続ケーブル添付)
PC-98H33	5インチ固定ディスクユニット	1ユニット目用，容量10Mバイト DMA転送方式による高速データ転送 (インタフェースボードとの接続ケーブル添付)
PC-98H34	拡張用5インチ固定ディスクユニット	2ユニット目用，容量10Mバイト DMA転送方式による高速データ転送 (接続ケーブル添付)

［プリンタ］

セントロニクス社準拠のインタフェースを採用しているPCシリーズのプリンタを接続することができます。

型　名	品　　名	特　　　長
PC-8024	ドットマトリックスプリンタ	9ピンドットマトリックスプリンタ 10インチ，印字速度　180CPS （接続ケーブル添付）
PC-8027	ドットマトリックスプリンタ	場所をとらないタテ型コンパクトタイプの9ピンドットマトリックスプリンタ 10インチ，印字速度　105CPS （接続ケーブル添付）
PC-8824	熱転写プリンタ	サーマルヘッドによるドットマトリックス方式 PC-8824-02を追加することによりPC-8825と同等 12インチ，印字速度　30CPS （接続ケーブル添付）
PC-8825	熱転写漢字プリンタ	明朝体（24ドット）の美しい漢字を印字（第一水準） PC-8824-12を追加することにより第二水準の漢字も印字可能 12インチ，印字速度　15CPS（漢字） （接続ケーブル添付）
PC-8826	カラープロッタプリンタ	4色マルチペンの採用で美しい絵を作図可能 プリンタ機能内蔵 10インチ （接続ケーブル添付）
PC-PR201	日本語シリアルプリンタ	明朝体（24ドット）の美しい漢字を印字可能（第一水準） PC-PR201-12を追加する事により第二水準の 漢字を印字可能 単票用紙の使用に便利なシートフィーダをオプションで用意 （PC-PR201-04） 16インチ，印字速度　40CPS（漢字） （接続ケーブル添付）
PC-PR202	ドットマトリックス漢字プリンタ	18ピンドットマトリックスプリンタ 漢字（16ドット）印字可能 15インチ，印字速度　60CPS（漢字） （接続ケーブル添付）

注　意

PC-8826では，PC8801MKⅡのCOPY機能は使用出来ません。

[拡張用ボード]

PC-8801MKⅡには3つの拡張スロットが用意されていますので，PC-8000シリーズやPC-8800シリーズ用に作られたインタフェースボードを使用する事ができます。

型名	品名	特長
PC-8012-01	ユニバーサルボード	オリジナル回路の作成用
PC-8012-02	増設RAMボード	32Kバイトのメモリボード 特定アプリケーションソフト上で使用
PC-8012-04	音声録音再生ボード	CVSD方式により自然な音声出力が可能 約30秒までの音声を記録
PC-8012-05	音声認識ボード	特定話者の音声を認識 128語まで登録可能
PC-8801-02N	増設RAMボード	128Kバイトメモリボード 特定アプリケーションソフト上で使用
PC-8801-07	5インチ・固定ディスクインタフェースボード	5インチ・固定ディスクを接続するためのインタフェースボード 固定ディスクはCP/MやPCNETの下で使用
PC-8801-10	ミュージックインタフェースボード	6重和音の音楽とシンセサイザーをコントロールするためのボード
PC-8801-16	16ビットカード	16ビット用のOS(MS-DOS)を使用するためのボード
PC-8863	専用通信ボード	エミュレータソフトを使用するためのボード
PC-8864	ネットワークインタフェースボード	PCNETを使用するためのボード
PC-8897	GP-IBインタフェースボード	GP-IB(IEEE-488)インタフェースを使用するためのボード

[入出力装置]

PC-8801MKⅡには，利用範囲をさらに広げる入出力装置が用意されています。

型名	品名	特長
PC-8268	パーソナルカプラ	電池駆動が可能な音響カプラ 300BPS用
PC-8269	インテリジェントテレホン	モデムを内蔵した電話機 300BPS用 (接続ケーブル添付)
PC-8875	パーソナルタブレット	専用ペンを使い0.1mmの分解能で座標を入力する装置 有効画面　A4判

[ケーブル]

PC-8801MKⅡと各種周辺機器を接続するには，つぎのケーブルを使用します。

型名	品名	特長
PC-8091	カラーディスプレイ用ケーブル	R. G. B信号入力型のカラーディスプレイを接続するためのケーブル
PC-8092	モノクロディスプレイ用ケーブル	コンポジット信号入力型のモノクロディスプレイを接続するためのケーブル (本体添付)
PC-8093	CMT用ケーブル	データレコーダを接続するためのケーブル (本体添付)
PC-8894	プリンタ用ケーブル	プリンタを接続するためのケーブル
PC-8895 PC-6093	RS-232C用ケーブル	RS-232Cインタフェースを接続するためのケーブル（モデム接続用）
PC-8896	GP-IB用ケーブル	GP-IBインタフェースを接続するためのケーブル
PC-CA601	RS-232C用ケーブル (ノーマル)	RS-232Cインタフェースを接続するためのケーブル（モデム接続用）
PC-CA602	RS-232C用ケーブル (リバース)	RS-232Cインタフェースを接続するためのケーブル（パソコン相互の接続用）

PC-8801MKⅡに接続出来るが注意が必要なもの

［ディスプレイ］

PC-8000シリーズ等の標準ディスプレイ(水平走査周波数15.75kHz)は640×200ドットモードで使用できます。

型名	品名	特長
PC-8044K	家庭テレビ用カラーアダプタ	このアダプタの接続により，家庭用カラーテレビを使用することが出来ます。解像度が低いので，細かい文字の表示にはむきません。
PC-8045K	ライトペン	キャラクタ単位の座標を読み取ります。ドット単位では使用できません。
PC-8048N	12インチ・カラーディスプレイ	家庭用テレビブラウン管(ドットピッチ0.63mm)を使用しているディスプレイ 接続ケーブル添付
PC-8049N	12インチ・カラーディスプレイ(高解像度)	高解像度ブラウン管(ドットピッチ0.31mm)を使用したカラーディスプレイ 接続ケーブル添付
PC-8050N	12インチ・グリーンディスプレイ	2000文字表示可能 接続ケーブルは本体添付
PC-8052	14インチ・カラーディスプレイ	ドットピッチ0.39mmのブラウン管を使用したカラーディスプレイ 接続ケーブル添付
PC-8054	14インチ・カラーディスプレイ	ドットピッチ0.50mmのブラウン管を使用したカラーディスプレイ 接続ケーブル添付
PC-8058	12インチ・カラーディスプレイ	ドットピッチ0.38mmのブラウン管を使用したカラーディスプレイ 接続ケーブル添付
PC-KD301	12インチ・カラーディスプレイ	ドットピッチ0.36mmのブラウン管を使用したカラーディスプレイ 接続ケーブル添付
PC-KD351	14インチ・カラーディスプレイ	ドットピッチ0.39mmのブラウン管を使用したカラーディスプレイ 接続ケーブル添付
PC-KD901	ディスプレイ置き台	PC-8853N, PC-KD551, PC-8054, PC-KD351専用ディスプレイ置き台

[プリンタ]

PC-2000シリーズ，PC-6000シリーズのプリンタを接続することが出来ますがCOPY機能は使用できません。

又、グラフィックキャラクタが異なります。

型名	品名	特長
PC-2021	サーマルプリンタ	電池駆動方式の40桁サーマルプリンタ 接続ケーブル(PC-6094-60)別売
PC-6021	40桁サーマルプリンタ	40桁サーマルプリンタ 接続ケーブル(PC-8894)別売
PC-6022	カラープロッタプリンタ	水性ボールペンタイプの4色カラープロッタプリンタ 接続ケーブル(PC-8894)別売
PC-6023	カラープロッタプリンタ	4色マルチペン(ボールペン/油性サインペン)を使用したA4判プロッタプリンタ 接続ケーブル(PC-8894)別売

付録7　漢字コード表

※コードは16進で表現されています。

　例えば"B"のコードは2340＋２＝2342となります。

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
半角文字	0020		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
	0030	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
	0040	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	0050	P	Q	R	S	T	U	V	W	X	Y	Z	[	¥	]	^	—
	0060		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	0070	p	q	r	s	t	u	v	w	x	y	z	{	¦	}	~	
	0080		。	「	」	、	・	を	ぁ	ぃ	ぅ	ぇ	ぉ	ゃ	ゅ	ょ	っ
	0090	ー	あ	い	う	え	お	か	き	く	け	こ	さ	し	す	せ	そ
	00A0		｡	｢	｣		･	ｦ	ｧ	ｨ	ｩ	ｪ	ｫ	ｬ	ｭ	ｮ	ｯ
	00B0	ｰ	ｱ	ｲ	ｳ	ｴ	ｵ	ｶ	ｷ	ｸ	ｹ	ｺ	ｻ	ｼ	ｽ	ｾ	ｿ
	00C0	ﾀ	ﾁ	ﾂ	ﾃ	ﾄ	ﾅ	ﾆ	ﾇ	ﾈ	ﾉ	ﾊ	ﾋ	ﾌ	ﾍ	ﾎ	ﾏ
	00D0	ﾐ	ﾑ	ﾒ	ﾓ	ﾔ	ﾕ	ﾖ	ﾗ	ﾘ	ﾙ	ﾚ	ﾛ	ﾜ	ﾝ	ﾞ	ﾟ
	00E0	た	ち	つ	て	と	な	に	ぬ	ね	の	は	ひ	ふ	へ	ほ	ま
	00F0	み	む	め	も	や	ゆ	よ	ら	り	る	れ	ろ	わ	ん	゛	゜
¼角文字	0100		S_H	S_X	E_X	E_T	E_Q	A_K	B_L	B_S	H_T	L_F	H_M	C_L	C_R	S_O	S_I
	0110	D_E	D_1	D_2	D_3	D_4	N_K	S_N	E_B	C_N	E_M	S_B	E_C	→	←	↑	↓
	0120		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
	0130	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
	0140	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	0150	P	Q	R	S	T	U	V	W	X	Y	Z	[	¥	]	^	—
	0160		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	0170	p	q	r	s	t	u	v	w	x	y	z	{	¦	}	~	
	0180	▁	▂	▃	▄	▅	▆	▇	█	▏	▎	▍	▌	▋	▊	▉	＋
	0190	┴	┬	┤	├	▔	─	▕	▕	┌	┐	└	┘	╭	╮	╰	╯
	01A0		｡	｢	｣	､	･	ｦ	ｧ	ｨ	ｩ	ｪ	ｫ	ｬ	ｭ	ｮ	ｯ
	01B0	ｰ	ｱ	ｲ	ｳ	ｴ	ｵ	ｶ	ｷ	ｸ	ｹ	ｺ	ｻ	ｼ	ｽ	ｾ	ｿ
	01C0	ﾀ	ﾁ	ﾂ	ﾃ	ﾄ	ﾅ	ﾆ	ﾇ	ﾈ	ﾉ	ﾊ	ﾋ	ﾌ	ﾍ	ﾎ	ﾏ
	01D0	ﾐ	ﾑ	ﾒ	ﾓ	ﾔ	ﾕ	ﾖ	ﾗ	ﾘ	ﾙ	ﾚ	ﾛ	ﾜ	ﾝ	ﾞ	ﾟ
	01E0	═	╞	╪	╡	◢	◣	◥	◤	♠	♥	♦	♣	●	○	╱	╲
	01F0	×	円	年	月	日	時	分	秒								
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
記号	2120			、	。	，	．	・	：	；	？	！	゛	゜	´	｀	¨
	2130	＾	￣	＿	ヽ	ヾ	ゝ	ゞ	〃	仝	々	〆	〇	ー	―	‐	／
	2140	＼	～	∥	｜	…	‥	‘	’	“	”	（	）	〔	〕	［	］
	2150	｛	｝	〈	〉	《	》	「	」	『	』	【	】	＋	－	±	×
	2160	÷	＝	≠	＜	＞	≦	≧	∞	∴	♂	♀	°	′	″	℃	￥
	2170	＄	￠	￡	％	＃	＆	＊	＠	§	☆	★	○	●	◎	◇	
	2220		◆	□	■	△	▲	▽	▼	※	〒	→	←	↑	↓	〓	
英・数字	2330	0	1	2	3	4	5	6	7	8	9						
	2340		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	2350	P	Q	R	S	T	U	V	W	X	Y	Z					
	2360		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	2370	p	q	r	s	t	u	v	w	x	y	z					
ひらがな	2420		ぁ	あ	ぃ	い	ぅ	う	ぇ	え	ぉ	お	か	が	き	ぎ	く
	2430	ぐ	け	げ	こ	ご	さ	ざ	し	じ	す	ず	せ	ぜ	そ	ぞ	た
	2440	だ	ち	ぢ	っ	つ	づ	て	で	と	ど	な	に	ぬ	ね	の	は
	2450	ば	ぱ	ひ	び	ぴ	ふ	ぶ	ぷ	へ	べ	ぺ	ほ	ぼ	ぽ	ま	み
	2460	む	め	も	ゃ	や	ゅ	ゆ	ょ	よ	ら	り	る	れ	ろ	ゎ	わ
	2470	ゐ	ゑ	を	ん												
カタカナ	2520		ァ	ア	ィ	イ	ゥ	ウ	ェ	エ	ォ	オ	カ	ガ	キ	ギ	ク
	2530	グ	ケ	ゲ	コ	ゴ	サ	ザ	シ	ジ	ス	ズ	セ	ゼ	ソ	ゾ	タ
	2540	ダ	チ	ヂ	ッ	ツ	ヅ	テ	デ	ト	ド	ナ	ニ	ヌ	ネ	ノ	ハ
	2550	バ	パ	ヒ	ビ	ピ	フ	ブ	プ	ヘ	ベ	ペ	ホ	ボ	ポ	マ	ミ
	2560	ム	メ	モ	ャ	ヤ	ュ	ユ	ョ	ヨ	ラ	リ	ル	レ	ロ	ヮ	ワ
	2570	ヰ	ヱ	ヲ	ン	ヴ	ヵ	ヶ									
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

※空白文字のコードは，2121Hです。

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ギリシア文字	2620		Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
	2630	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω							
	2640		α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
	2650	π	ρ	σ	τ	υ	φ	χ	ψ	ω							
ロシア文字	2720		А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н
	2730	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э
	2740	Ю	Я														
	2750		а	б	в	г	д	е	ё	ж	з	и	й	к	л	м	н
	2760	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э
	2770	ю	я														
ア	3020		亜	唖	娃	阿	哀	愛	挨	姶	逢	葵	茜	穐	悪	握	渥
	3030	旭	葦	芦	鯵	梓	圧	斡	扱	宛	姐	虻	飴	絢	綾	鮎	或
	3040	粟	袷	安	庵	按	暗	案	闇	鞍	杏						
イ	3040											以	伊	位	依	偉	囲
	3050	夷	委	威	尉	惟	意	慰	易	椅	為	畏	異	移	維	緯	胃
	3060	萎	衣	謂	違	遺	医	井	亥	域	育	郁	磯	一	壱	溢	逸
	3070	稲	茨	芋	鰯	允	印	咽	員	因	姻	引	飲	淫	胤	蔭	
	3120		院	陰	隠	韻	吋										
ウ	3120							右	宇	烏	羽	迂	雨	卯	鵜	窺	丑
	3130	碓	臼	渦	嘘	唄	欝	蔚	鰻	姥	厩	浦	瓜	閏	噂	云	運
	3140	雲															
エ	3140		荏	餌	叡	営	嬰	影	映	曳	栄	永	泳	洩	瑛	盈	穎
	3150	頴	英	衛	詠	鋭	液	疫	益	駅	悦	謁	越	閲	榎	厭	円
	3160	園	堰	奄	宴	延	怨	掩	援	沿	演	炎	焔	煙	燕	猿	縁
	3170	艶	苑	薗	遠	鉛	鴛	塩									
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
オ	3170								於	汚	甥	凹	央	奥	往	応	
	3220		押	旺	横	欧	殴	王	翁	襖	鶯	鴎	黄	岡	沖	荻	億
	3230	屋	憶	臆	桶	牡	乙	俺	卸	恩	温	穏	音				
カ	3230													下	化	仮	何
	3240	伽	価	佳	加	可	嘉	夏	嫁	家	寡	科	暇	果	架	歌	河
	3250	火	珂	禍	禾	稼	箇	花	苛	茄	荷	華	菓	蝦	課	嘩	貨
	3260	迦	過	霞	蚊	俄	峨	我	牙	画	臥	芽	蛾	賀	雅	餓	駕
	3270	介	会	解	回	塊	壊	廻	快	怪	悔	恢	懐	戒	拐	改	
	3320		魁	晦	械	海	灰	界	皆	絵	芥	蟹	開	階	貝	凱	劾
	3330	外	咳	害	崖	慨	概	涯	碍	蓋	街	該	鎧	骸	浬	馨	蛙
	3340	垣	柿	蠣	鈎	劃	嚇	各	廓	拡	攪	格	核	殻	獲	確	穫
	3350	覚	角	赫	較	郭	閣	隔	革	学	岳	楽	額	顎	掛	笠	樫
	3360	橿	梶	鰍	潟	割	喝	恰	括	活	渇	滑	葛	褐	轄	且	鰹
	3370	叶	椛	樺	鞄	株	兜	竈	蒲	釜	鎌	噛	鴨	栢	茅	萱	
	3420		粥	刈	苅	瓦	乾	侃	冠	寒	刊	勘	勧	巻	喚	堪	姦
	3430	完	官	寛	干	幹	患	感	慣	憾	換	敢	柑	桓	棺	款	歓
	3440	汗	漢	澗	灌	環	甘	監	看	竿	管	簡	緩	缶	翰	肝	艦
	3450	莞	観	諫	貫	還	鑑	間	閑	関	陥	韓	館	舘	丸	含	岸
	3460	巌	玩	癌	眼	岩	翫	贋	雁	頑	顔	願					
キ	3460												企	伎	危	喜	器
	3470	基	奇	嬉	寄	岐	希	幾	忌	揮	机	旗	既	期	棋	棄	
	3520		機	帰	毅	気	汽	畿	祈	季	稀	紀	徽	規	記	貴	起
	3530	軌	輝	飢	騎	鬼	亀	偽	儀	妓	宜	戯	技	擬	欺	犠	疑
	3540	祇	義	蟻	誼	議	掬	菊	鞠	吉	吃	喫	桔	橘	詰	砧	杵
	3550	黍	却	客	脚	虐	逆	丘	久	仇	休	及	吸	宮	弓	急	救
	3560	朽	求	汲	泣	灸	球	究	窮	笈	級	糾	給	旧	牛	去	居
	3570	巨	拒	拠	挙	渠	虚	許	距	鋸	漁	禦	魚	亨	享	京	
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	3620		供	侠	僑	兇	競	共	凶	協	匡	卿	叫	喬	境	峡	強
	3630	彊	怯	恐	恭	挾	教	橋	況	狂	狭	矯	胸	脅	興	蕎	郷
	3640	鏡	響	饗	驚	仰	凝	尭	暁	業	局	曲	極	玉	桐	粁	僅
	3650	勤	均	巾	錦	斤	欣	欽	琴	禁	禽	筋	緊	芹	菌	衿	襟
	3660	謹	近	金	吟	銀											
ク	3660						九	倶	句	区	狗	玖	矩	苦	躯	駆	駈
	3670	駒	具	愚	虞	喰	空	偶	寓	遇	隅	串	櫛	釧	屑	屈	
	3720		掘	窟	沓	靴	轡	窪	熊	隈	粂	栗	繰	桑	鍬	勲	君
	3730	薫	訓	群	軍	郡											
ケ	3730						卦	袈	祁	係	傾	刑	兄	啓	圭	珪	型
	3740	契	形	径	恵	慶	慧	憩	掲	携	敬	景	桂	渓	畦	稽	系
	3750	経	継	繋	罫	茎	荊	蛍	計	詣	警	軽	頚	鶏	芸	迎	鯨
	3760	劇	戟	撃	激	隙	桁	傑	欠	決	潔	穴	結	血	訣	月	件
	3770	倹	倦	健	兼	券	剣	喧	圏	堅	嫌	建	憲	懸	拳	捲	
	3820		検	権	牽	犬	献	研	硯	絹	県	肩	見	謙	賢	軒	遣
	3830	鍵	険	顕	験	鹸	元	原	厳	幻	弦	減	源	玄	現	絃	舷
	3840	言	諺	限													
コ	3840				乎	個	古	呼	固	姑	孤	己	庫	弧	戸	故	枯
	3850	湖	狐	糊	袴	股	胡	菰	虎	誇	跨	鈷	雇	顧	鼓	五	互
	3860	伍	午	呉	吾	娯	後	御	悟	梧	檎	瑚	碁	語	誤	護	醐
	3870	乞	鯉	交	佼	侯	候	倖	光	公	功	効	勾	厚	口	向	
	3920		后	喉	坑	垢	好	孔	孝	宏	工	巧	巷	幸	広	庚	康
	3930	弘	恒	慌	抗	拘	控	攻	昂	晃	更	杭	校	梗	構	江	洪
	3940	浩	港	溝	甲	皇	硬	稿	糠	紅	紘	絞	綱	耕	考	肯	肱
	3950	腔	膏	航	荒	行	衡	講	貢	購	郊	酵	鉱	砿	鋼	閤	降
	3960	項	香	高	鴻	剛	劫	号	合	壕	拷	濠	豪	轟	麹	克	刻
	3970	告	国	穀	酷	鵠	黒	獄	漉	腰	甑	忽	惚	骨	狛	込	
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	3A20		此	頃	今	困	坤	墾	婚	恨	懇	昏	昆	根	梱	混	痕
	3A30	紺	艮	魂													
サ	3A30				些	佐	叉	唆	嵯	左	差	査	沙	瑳	砂	詐	鎖
	3A40	裟	坐	座	挫	債	催	再	最	哉	塞	妻	宰	彩	才	採	栽
	3A50	歳	済	災	采	犀	砕	砦	祭	斎	細	菜	裁	載	際	剤	在
	3A60	材	罪	財	冴	坂	阪	堺	榊	肴	咲	崎	埼	碕	鷺	作	削
	3A70	咋	搾	昨	朔	柵	窄	策	索	錯	桜	鮭	笹	匙	冊	刷	
	3B20		察	拶	撮	擦	札	殺	薩	雑	皐	鯖	捌	錆	鮫	皿	晒
	3B30	三	傘	参	山	惨	撒	散	桟	燦	珊	産	算	纂	蚕	讃	賛
	3B40	酸	餐	斬	暫	残											
シ	3B40						仕	仔	伺	使	刺	司	史	嗣	四	士	始
	3B50	姉	姿	子	屍	市	師	志	思	指	支	孜	斯	施	旨	枝	止
	3B60	死	氏	獅	祉	私	糸	紙	紫	肢	脂	至	視	詞	詩	試	誌
	3B70	諮	資	賜	雌	飼	歯	事	似	侍	児	字	寺	慈	持	時	
	3C20		次	滋	治	爾	璽	痔	磁	示	而	耳	自	蒔	辞	汐	鹿
	3C30	式	識	鴫	竺	軸	宍	雫	七	叱	執	失	嫉	室	悉	湿	漆
	3C40	疾	質	実	蔀	篠	偲	柴	芝	屡	蕊	縞	舎	写	射	捨	赦
	3C50	斜	煮	社	紗	者	謝	車	遮	蛇	邪	借	勺	尺	杓	灼	爵
	3C60	酌	釈	錫	若	寂	弱	惹	主	取	守	手	朱	殊	狩	珠	種
	3C70	腫	趣	酒	首	儒	受	呪	寿	授	樹	綬	需	囚	収	周	
	3D20		宗	就	州	修	愁	拾	洲	秀	秋	終	繍	習	臭	舟	蒐
	3D30	衆	襲	讐	蹴	輯	週	酋	酬	集	醜	什	住	充	十	従	戎
	3D40	柔	汁	渋	獣	縦	重	銃	叔	夙	宿	淑	祝	縮	粛	塾	熟
	3D50	出	術	述	俊	峻	春	瞬	竣	舜	駿	准	循	旬	楯	殉	淳
	3D60	準	潤	盾	純	巡	遵	醇	順	処	初	所	暑	曙	渚	庶	緒
	3D70	署	書	薯	藷	諸	助	叙	女	序	徐	恕	鋤	除	傷	償	
	3E20		勝	匠	升	召	哨	商	唱	嘗	奨	妾	娼	宵	将	小	少
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	3E30	尚	庄	床	廠	彰	承	抄	招	掌	捷	昇	昌	昭	晶	松	梢
	3E40	樟	樵	沼	消	渉	湘	焼	焦	照	症	省	硝	礁	祥	称	章
	3E50	笑	粧	紹	肖	菖	蒋	蕉	衝	裳	訟	証	詔	詳	象	賞	醤
	3E60	鉦	鍾	鐘	障	鞘	上	丈	丞	乗	冗	剰	城	場	壌	嬢	常
	3E70	情	擾	条	杖	浄	状	畳	穣	蒸	譲	醸	錠	嘱	埴	飾	
	3F20		拭	植	殖	燭	織	職	色	触	食	蝕	辱	尻	伸	信	侵
	3F30	唇	娠	寝	審	心	慎	振	新	晋	森	榛	浸	深	申	疹	真
	3F40	神	秦	紳	臣	芯	薪	親	診	身	辛	進	針	震	人	仁	刃
	3F50	塵	壬	尋	甚	尽	腎	訊	迅	陣	靭						
ス	3F50											笥	諏	須	酢	図	厨
	3F60	逗	吹	垂	帥	推	水	炊	睡	粋	翠	衰	遂	酔	錐	錘	随
	3F70	瑞	髄	崇	嵩	数	枢	趨	雛	据	杉	椙	菅	頗	雀	裾	
	4020		澄	摺	寸												
セ	4020					世	瀬	畝	是	凄	制	勢	姓	征	性	成	政
	4030	整	星	晴	棲	栖	正	清	牲	生	盛	精	聖	声	製	西	誠
	4040	誓	請	逝	醒	青	静	斉	税	脆	隻	席	惜	戚	斥	昔	析
	4050	石	積	籍	績	脊	責	赤	跡	蹟	碩	切	拙	接	摂	折	設
	4060	窃	節	説	雪	絶	舌	蝉	仙	先	千	占	宣	専	尖	川	戦
	4070	扇	撰	栓	栴	泉	浅	洗	染	潜	煎	煽	旋	穿	箭	線	
	4120		繊	羨	腺	舛	船	薦	詮	賎	践	選	遷	銭	銑	閃	鮮
	4130	前	善	漸	然	全	禅	繕	膳	糎							
ソ	4130										噌	塑	岨	措	曾	曽	楚
	4140	狙	疏	疎	礎	祖	租	粗	素	組	蘇	訴	阻	遡	鼠	僧	創
	4150	双	叢	倉	喪	壮	奏	爽	宋	層	匝	惣	想	捜	掃	挿	掻
	4160	操	早	曹	巣	槍	槽	漕	燥	争	痩	相	窓	糟	総	綜	聡
	4170	草	荘	葬	蒼	藻	装	走	送	遭	鎗	霜	騒	像	増	憎	
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	4220		臓	蔵	贈	造	促	側	則	即	息	捉	束	測	足	速	俗
	4230	属	賊	族	続	卒	袖	其	揃	存	孫	尊	損	村	遜		
タ	4230															他	多
	4240	太	汰	詑	唾	堕	妥	惰	打	柁	舵	楕	陀	駄	騨	体	堆
	4250	対	耐	岱	帯	待	怠	態	戴	替	泰	滞	胎	腿	苔	袋	貸
	4260	退	逮	隊	黛	鯛	代	台	大	第	醍	題	鷹	滝	瀧	卓	啄
	4270	宅	托	択	拓	沢	濯	琢	託	鐸	濁	諾	茸	凧	蛸	只	
	4320		叩	但	達	辰	奪	脱	巽	竪	辿	棚	谷	狸	鱈	樽	誰
	4330	丹	単	嘆	坦	担	探	旦	歎	淡	湛	炭	短	端	箪	綻	耽
	4340	胆	蛋	誕	鍛	団	壇	弾	断	暖	檀	段	男	談			
チ	4340														値	知	地
	4350	弛	恥	智	池	痴	稚	置	致	蜘	遅	馳	築	畜	竹	筑	蓄
	4360	逐	秩	窒	茶	嫡	着	中	仲	宙	忠	抽	昼	柱	注	虫	衷
	4370	註	酎	鋳	駐	樗	瀦	猪	苧	著	貯	丁	兆	凋	喋	寵	
	4420		帖	帳	庁	弔	張	彫	徴	懲	挑	暢	朝	潮	牒	町	眺
	4430	聴	脹	腸	蝶	調	諜	超	跳	銚	長	頂	鳥	勅	捗	直	朕
	4440	沈	珍	賃	鎮	陳											
ツ	4440						津	墜	椎	槌	追	鎚	痛	通	塚	栂	掴
	4450	槻	佃	漬	柘	辻	蔦	綴	鍔	椿	潰	坪	壷	嬬	紬	爪	吊
	4460	釣	鶴														
テ	4460			亭	低	停	偵	剃	貞	呈	堤	定	帝	底	庭	廷	弟
	4470	悌	抵	挺	提	梯	汀	碇	禎	程	締	艇	訂	諦	蹄	逓	
	4520		邸	鄭	釘	鼎	泥	摘	擢	敵	滴	的	笛	適	鏑	溺	哲
	4530	徹	撤	轍	迭	鉄	典	填	天	展	店	添	纏	甜	貼	転	顛
	4540	点	伝	殿	澱	田	電										
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ト	4540							兎	吐	堵	塗	妬	屠	徒	斗	杜	渡
	4550	登	菟	賭	途	都	鍍	砥	礪	努	度	土	奴	怒	倒	党	冬
	4560	凍	刀	唐	塔	塘	套	宕	島	嶋	悼	投	搭	東	桃	檮	棟
	4570	盗	淘	湯	濤	灯	燈	当	痘	禱	等	答	筒	糖	統	到	
	4620		董	蕩	藤	討	謄	豆	踏	逃	透	鐙	陶	頭	騰	闘	働
	4630	動	同	堂	導	憧	撞	洞	瞳	童	胴	萄	道	銅	峠	鴇	匿
	4640	得	徳	瀆	特	督	禿	篤	毒	独	読	栃	橡	凸	突	椴	届
	4650	鳶	苫	寅	酉	瀞	噸	屯	惇	敦	沌	豚	遁	頓	呑	曇	鈍
ナ	4660	奈	那	内	乍	凪	薙	謎	灘	捺	鍋	楢	馴	縄	畷	南	楠
	4670	軟	難	汝													
ニ	4670				二	尼	弐	迩	匂	賑	肉	虹	廿	日	乳	入	
	4720		如	尿	韮	任	妊	忍	認								
ヌ	4720									濡							
ネ	4720										禰	祢	寧	葱	猫	熱	年
	4730	念	捻	撚	燃	粘											
ノ	4730						乃	廼	之	埜	嚢	悩	濃	納	能	脳	膿
	4740	農	覗	蚤													
ハ	4740				巴	把	播	覇	杷	波	派	琶	破	婆	罵	芭	馬
	4750	俳	廃	拝	排	敗	杯	盃	牌	背	肺	輩	配	倍	培	媒	梅
	4760	楳	煤	狽	買	売	賠	陪	這	蝿	秤	矧	萩	伯	剥	博	拍
	4770	柏	泊	白	箔	粕	舶	薄	迫	曝	漠	爆	縛	莫	駁	麦	
	4820		函	箱	硲	箸	肇	筈	櫨	幡	肌	畑	畠	八	鉢	溌	発
	4830	醗	髪	伐	罰	抜	筏	閥	鳩	噺	塙	蛤	隼	伴	判	半	反
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	4840	叛	帆	搬	斑	板	氾	汎	版	犯	班	畔	繁	般	藩	販	範
	4850	釆	煩	頒	飯	挽	晩	番	盤	磐	蕃	蛮					
ヒ	4850												匪	卑	否	妃	庇
	4860	彼	悲	扉	批	披	斐	比	泌	疲	皮	碑	秘	緋	罷	肥	被
	4870	誹	費	避	非	飛	樋	簸	備	尾	微	枇	毘	琵	眉	美	
	4920		鼻	柊	稗	匹	疋	髭	彦	膝	菱	肘	弼	必	畢	筆	逼
	4930	桧	姫	媛	紐	百	謬	俵	彪	標	氷	漂	瓢	票	表	評	豹
	4940	廟	描	病	秒	苗	錨	鋲	蒜	蛭	鰭	品	彬	斌	浜	瀕	貧
	4950	賓	頻	敏	瓶												
フ	4950					不	付	埠	夫	婦	富	冨	布	府	怖	扶	敷
	4960	斧	普	浮	父	符	腐	膚	芙	譜	負	賦	赴	阜	附	侮	撫
	4970	武	舞	葡	蕪	部	封	楓	風	葺	蕗	伏	副	復	幅	服	
	4A20		福	腹	複	覆	淵	弗	払	沸	仏	物	鮒	分	吻	噴	墳
	4A30	憤	扮	焚	奮	粉	糞	紛	雰	文	聞						
ヘ	4A30											丙	併	兵	塀	幣	平
	4A40	弊	柄	並	蔽	閉	陛	米	頁	僻	壁	癖	碧	別	瞥	蔑	箆
	4A50	偏	変	片	篇	編	辺	返	遍	便	勉	娩	弁	鞭			
ホ	4A50														保	舗	鋪
	4A60	圃	捕	歩	甫	補	輔	穂	募	墓	慕	戊	暮	母	簿	菩	倣
	4A70	俸	包	呆	報	奉	宝	峰	峯	崩	庖	抱	捧	放	方	朋	
	4B20		法	泡	烹	砲	縫	胞	芳	萌	蓬	蜂	褒	訪	豊	邦	鋒
	4B30	飽	鳳	鵬	乏	亡	傍	剖	坊	妨	帽	忘	忙	房	暴	望	某
	4B40	棒	冒	紡	肪	膨	謀	貌	貿	鉾	防	吠	頬	北	僕	卜	墨
	4B50	撲	朴	牧	睦	穆	釦	勃	没	殆	堀	幌	奔	本	翻	凡	盆

0 1 2 3 4 5 6 7 8 9 A B C D E F

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
マ	4B60	摩	磨	魔	麻	埋	妹	昧	枚	毎	哩	槙	幕	膜	枕	鮪	柾
	4B70	鱒	桝	亦	俣	又	抹	末	沫	迄	儘	繭	麿	万	慢	満	
	4C20		漫	蔓													
ミ	4C20				味	未	魅	巳	箕	岬	密	蜜	湊	蓑	稔	脈	妙
	4C30	粍	民	眠													
ム	4C30				務	夢	無	牟	矛	霧	鵡	椋	婿	娘			
メ	4C30														冥	名	命
	4C40	明	盟	迷	銘	鳴	姪	牝	滅	免	棉	綿	緬	面	麺		
モ	4C40															摸	模
	4C50	茂	妄	孟	毛	猛	盲	網	耗	蒙	儲	木	黙	目	杢	勿	餅
	4C60	尤	戻	籾	貰	問	悶	紋	門	匁							
ヤ	4C60										也	冶	夜	爺	耶	野	弥
	4C70	矢	厄	役	約	薬	訳	躍	靖	柳	薮	鑓					
ユ	4C70												愉	愈	油	癒	
	4D20		諭	輸	唯	佑	優	勇	友	宥	幽	悠	憂	揖	有	柚	湧
	4D30	涌	猶	猷	由	祐	裕	誘	遊	邑	郵	雄	融	夕			
ヨ	4D30														予	余	与
	4D40	誉	輿	預	傭	幼	妖	容	庸	揚	揺	擁	曜	楊	様	洋	溶
	4D50	熔	用	窯	羊	耀	葉	蓉	要	謡	踊	遥	陽	養	慾	抑	欲
	4D60	沃	浴	翌	翼	淀											
ラ	4D60						羅	螺	裸	来	莱	頼	雷	洛	絡	落	酪
	4D70	乱	卵	嵐	欄	濫	藍	蘭	覧								
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
リ	4D70									利	吏	履	李	梨	理	璃	
	4E20		痢	裏	裡	里	離	陸	律	率	立	葎	掠	略	劉	流	溜
	4E30	琉	留	硫	粒	隆	竜	龍	侶	慮	旅	虜	了	亮	僚	両	凌
	4E40	寮	料	梁	涼	猟	療	瞭	稜	糧	良	諒	遼	量	陵	領	力
	4E50	緑	倫	厘	林	淋	燐	琳	臨	輪	隣	鱗	麟				
ル	4E50													瑠	塁	涙	累
	4E60	類															
レ	4E60		令	伶	例	冷	励	嶺	怜	玲	礼	苓	鈴	隷	零	霊	麗
	4E70	齢	暦	歴	列	劣	烈	裂	廉	恋	憐	漣	煉	簾	練	聯	
	4F20		蓮	連	錬												
ロ	4F20					呂	魯	櫓	炉	賂	路	露	労	婁	廊	弄	朗
	4F30	楼	榔	浪	漏	牢	狼	籠	老	聾	蠟	郎	六	麓	禄	肋	録
	4F40	論															
ワ	4F40		倭	和	話	歪	賄	脇	惑	枠	鷲	亙	亘	鰐	詫	藁	蕨
	4F50	椀	湾	碗	腕												
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

付録8　キャラクタコード表

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		D_E		0	@	P		p	▁	┴		ー	タ	ミ	═	╳
1	S_H	D_1	!	1	A	Q	a	q	▂	┬	。	ア	チ	ム	╞	円
2	S_X	D_2	"	2	B	R	b	r	▃	┤	「	イ	ツ	メ	╪	年
3	E_X	D_3	#	3	C	S	c	s	▄	├	」	ウ	テ	モ	╡	月
4	E_T	D_4	$	4	D	T	d	t	▅	▔	、	エ	ト	ヤ	◢	日
5	E_Q	N_K	%	5	E	U	e	u	▆	━	・	オ	ナ	ユ	◣	時
6	A_K	S_N	&	6	F	V	f	v	▇	▐	ヲ	カ	ニ	ヨ	◥	分
7	B_L	E_B	'	7	G	W	g	w	█	▕	ァ	キ	ヌ	ラ	◤	秒
8	B_S	C_N	(	8	H	X	h	x	▏	┌	ィ	ク	ネ	リ	♠	
9	H_T	E_M	)	9	I	Y	i	y	▎	┐	ゥ	ケ	ノ	ル	♥	
A	L_F	S_B	*	:	J	Z	j	z	▍	└	ェ	コ	ハ	レ	♦	
B	H_M	E_C	+	;	K	[	k	{	▌	┘	ォ	サ	ヒ	ロ	♣	
C	C_L	→	,	<	L	¥	l	¦	▋	╭	ャ	シ	フ	ワ	●	
D	C_R	←	−	=	M	]	m	}	▊	╮	ュ	ス	ヘ	ン	○	
E	S_O	↑	.	>	N	^	n	～	▉	╰	ョ	セ	ホ	゛	╱	
F	S_I	↓	/	?	O	_	o		┼	╯	ッ	ソ	マ	゜	╲	